On the Performance of Gradient Tracking with Local Updates

Anonymous Authors[1]

Abstract

We study the decentralized optimization problem where a network of n agents seeks to minimize the average of a set of heterogeneous non-convex cost functions distributedly. State-of-the-art decentralized algorithms like Exact Diffusion and Gradient Tracking (GT) involve communicating every iteration. However, communication is expensive, resource intensive, and slow. This work analyzes a locally updated GT method (LU-GT), where agents perform local recursions before interacting with their neighbors. While local updates have been shown to reduce communication overhead in practice, their theoretical influence has not been fully characterized. We show LU-GT has the same communication complexity as the Federated Learning setting but allows for decentralized (symmetric) network topologies and prove that the number of local updates does not degrade the quality of the solution achieved by LU-GT.

1. Introduction

We study the distributed multi-agent optimization problem

$$\underset{x\in\mathbb{R}^m}{\text{minimize}} \quad f(x) \triangleq \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{1}$$

where $f_i(\cdot) : \mathbb{R}^m \to \mathbb{R}$ is a smooth, *non-convex* function held privately by agent $i \in \{1, \ldots, n\}$. The agents collaborate to find a consensual solution x^* of (1) with communication constrained by some network topology.

Many decentralized methods have been proposed to solve (1). Among the most prolific include decentralized/distributed gradient descent (DGD) (Ram et al., 2010; Cattivelli & Sayed, 2010), EXTRA (Shi et al., 2015), Exact-Diffusion/D^2/NIDS (ED) (Yuan et al., 2019; Li et al., 2019; Yuan et al., 2020; Tang et al., 2018), and Gradient Tracking (GT) (Xu et al., 2015; Di Lorenzo & Scutari, 2016; Qu & Li, 2018; Nedic et al., 2017). DGD is an algorithm wherein agents perform a local gradient step followed by a communication round. However, DGD has been shown not optimal for constant stepsizes when agents' local objective functions are heterogeneous, i.e., the minimizer of functions $f_i(\cdot)$ differs from the minimizer of $f(\cdot)$. This shortcoming has been analyzed in (Chen & Sayed, 2013; Yuan et al., 2016) where the heterogeneity causes the rate of DGD to incur an additional bias term with a magnitude directly proportional to the level of heterogeneity. Moreover, this bias term is inversely influenced by the connectivity of the network (becomes larger for sparse networks) (Yuan et al., 2020; Koloskova et al., 2020).

EXTRA, ED, and GT employ bias-correction techniques to account for heterogeneity. EXTRA and ED use local updates that incorporate the previous iteration's parameter and gradient. GT methods have each agent perform the local update with an estimate of the global gradient called the tracking variable. In these techniques, the bias term proportional to the heterogeneity found in DGD is removed (Alghunaim & Yuan, 2022; Koloskova et al., 2021). However, they require communication over the network at every iteration.

Communication is expensive, resource intensive, and slow in practice (Ying et al., 2021). Centralized methods in which agents communicate with a central coordinator (i.e., server) have been developed to solve (1) with an explicit focus on reducing the communication cost. This has been achieved empirically by requiring agents to perform local recursions before communicating. Among these methods include LocalGD (Stich, 2019; Khaled et al., 2019; 2020b; Zhang et al., 2016; Lin et al., 2020), Scaffold (Karimireddy et al., 2020), S-Local-GD (Gorbunov et al., 2021), FedLin (Mitra et al., 2021), and Scaffnew (Mishchenko et al., 2022). Analysis on LocalGD revealed that local recursions cause agents to drift towards their local solution (Khaled et al., 2019; 2020a; Koloskova et al., 2020). Scaffold, S-Local-GD, FedLin, and Scaffnew address this issue by introducing bias-correction techniques. However, besides (Mishchenko et al., 2022), analysis of these methods has failed to show communication complexity improvements. The work (Mishchenko et al., 2022) has shown that for μ-strongly-convex, L-smooth, and

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

deterministic functions, the communication complexity of Scaffnew can be improved from $O(\kappa)$ to $O(\sqrt{\kappa})$ if one performs $\sqrt{\kappa}$ local recursions with $\kappa \triangleq L/\mu$.

Local recursions in *decentralized methods* have been much less studied. DGD with local recursions has been studied in (Koloskova et al., 2020), but the convergence rates still have bias terms due to heterogeneity. Additionally, the magnitude of the bias term is proportional to the number of local recursions taken. Scaffnew (Mishchenko et al., 2022) has been studied under the decentralized case but for the strongly convex and smooth function class. In (Mishchenko et al., 2022), for sufficiently connected graphs, an improvement to a communication complexity of $O(\sqrt{\kappa/(1-\lambda)})$ where λ is the mixing rate of the matrix is shown. Several works studied GT under time-varying graphs such as (Di Lorenzo & Scutari, 2016; Nedic et al., 2017; Scutari & Sun, 2019; Sun et al., 2022; Saadatniaki et al., 2020), among these only the works (Di Lorenzo & Scutari, 2016; Scutari & Sun, 2019; Lu & Wu, 2020) considered nonconvex setting. Different from (Di Lorenzo & Scutari, 2016; Scutari & Sun, 2019; Lu & Wu, 2020), we provide explicit expressions that characterize the convergence rate in terms of the problem parameters (e.g., network topology).

In this work, we propose and study LU-GT, a locally updated decentralized algorithm based on the bias-corrected method GT. Our contributions are as follows:

- We analyze LU-GT under the deterministic, nonconvex regime. As a byproduct, we provide an alternative and simpler analysis for GT, which extends the techniques from (Alghunaim & Yuan, 2022).
- We show LU-GT has a communication complexity matching locally updated variants of federated algorithms.
- We demonstrate that LU-GT retains the bias-correction properties of GT irrespective of the number of local recursions and that the number of local recursions does not affect the quality of the solution.
- Numerical analysis shows that local recursions can reduce the communication overhead in certain regimes, e.g., well-connected graphs.

This paper is organized as follows. Section 2 defines relevant notation, states the assumptions used in our analysis, introduces LU-GT, and states our main result on the convergence rate. In Section A, we provide intuition into how the direction of our analysis can show that following LU-GT, agents reach a consensus that is also a first-order stationary point. We also cover relevant lemmas needed in the analysis of LU-GT. In Section B, we prove the convergence rate of LU-GT. Section 3 shows evidence that the local recursions of LU-GT can reduce communication costs in certain regimes.

Notation: Lowercase letters define vectors or scalars, while uppercase letters define matrices. We let $\mathrm{col}\{a_1, ..., a_n\}$ or $\mathrm{col}\{a_i\}_{i=1}^n$ denote the vector that concatenates the vectors/scalars a_i. We let $\mathrm{diag}\{d_1, ..., d_n\}$ or $\mathrm{diag}\{d_i\}_{i=1}^n$ denote the matrix with diagonal elements d_i. Similarly, blkdiag$\{D_1, D_2, ..., D_n\}$ or blkdiag$\{D_i\}_{i=1}^n$ represents the block diagonal matrix with matrices D_i along the diagonal. The notation $\mathbf{1}$ represents the one vector of size that should be inferred while $\mathbf{1}_n$ represents the one vector of size n. The inner product of two vectors a, b is defined as $\langle a, b\rangle$. $\otimes$ represents the Kronecker product. Boldface variables such as $(\mathbf{x}, \mathbf{W})$ represent augmented network quantities.

2. Algorithm, Assumptions, and Main Result

The original gradient tracking method has the form (Xu et al., 2015):

$$x_i^{k+1} = \sum_{j\in\mathcal{N}_i} w_{ij}(x_j^k - \bar{\eta} g_j^k) \tag{2a}$$

$$g_i^{k+1} = \sum_{j\in\mathcal{N}_i} w_{ij}\big(g_j^k + \nabla f_j(x_j^{k+1}) - \nabla f_j(x_j^k)\big), \tag{2b}$$

with $g_i^0 = \nabla f(x_i^0)$. Here, x_i^k is agent i's current parameter estimate at iteration k, and $g_i^k \in \mathbb{R}^n$ is an additional parameter held by agent i that tracks the average of the gradient. Here, w_{ij} is a scalar weight that scales the information agent i receives from agent j, and $\mathcal{N}_i$ is the set of neighbors of agent i. We set $w_{ij} = 0$ if $j \notin \mathcal{N}_i$.

In this work, we study a locally updated variant of gradient tracking listed in Algorithm 1 where instead of agents communicating every iteration, they communicate every T_{o} iterations. The proposed method LU-GT is detailed in Algorithm 1 where α and η are step-size parameters, and T_{o} is the number of local recursions before a round of communication. The intuition behind the algorithm is to have agents perform a descent step using a staling estimate of the global gradient for T_{o} iterations. Afterwards, agents perform a weighted average of their parameters with their neighbors and update their tracking variable.

Remark 2.1. For $T_{\mathrm{o}} = 1$, Algorithm 1 becomes equivalent to the original ATC-GT (Xu et al., 2015) with stepsize $\bar{\eta} = \eta\alpha$. This can be seen by introducing the change of variable $g_i^k = (1/\alpha)y_i^k$. Thus, our analysis also covers the original GT method.

For analysis reasons, we will rewrite algorithm 1 using network notation. To do so,we define $W = [w_{ij}] \in \mathbb{R}^{n\times n}$ as the mixing matrix for an undirected graph that models the connections of a group of n agents. We also introduce

Algorithm 1 LU-GT for each agent i

```
Input: $x_i^0 = 0 \in \mathbb{R}^m$, $y_i^0 = \alpha\nabla f_i(x_i^0)$, $\alpha > 0$, $\eta > 0$
    $T_o \in \mathbb{Z}_{\geq 0}$, $K \in \mathbb{Z}_+$
Define: $\tau = \{0, T_o, 2T_o, 3T_o ...\}$
for $k = 0, ..., K-1$ do
    if $k \in \tau$ then
        $x_i^{k+1} = \sum_{j\in\mathcal{N}_i} w_{ij}(x_j^k - \eta y_j^k)$
        $y_i^{k+1} = \sum_{j\in\mathcal{N}_i} w_{ij}(y_j^k + \alpha\nabla f_j(x_j^{k+1}) - \alpha\nabla f_j(x_j^k))$
    else
        $x_i^{k+1} = x_i^k - \eta y_i^k$
        $y_i^{k+1} = y_i^k + \alpha\nabla f_i(x_i^{k+1}) - \alpha\nabla f_i(x_i^k)$
    end if
end for
```

the network notations:

$$\mathbf{W} = W \otimes I_d \in \mathbb{R}^{mn\times mn}$$

$$\mathbf{x}^k = \mathrm{col}\{x_1^k, \dots, x_n^k\}, \quad \mathbf{y}^k = \mathrm{col}\{y_1^k, \dots, y_n^k\}$$

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^n f_i(x_i), \;\; \nabla\mathbf{f}(\mathbf{x}) = \mathrm{col}\{\nabla f_1(x_1), \dots, \nabla f_n(x_n)\}.$$

To analyze Algorithm 1, we first introduce the following time-varying matrix:

$$\mathbf{W}_k \triangleq \begin{cases} \mathbf{W} & \text{when } k \in \tau, \\ \mathbf{I} & \text{otherwise.} \end{cases} \tag{4}$$

Thus, we can succinctly rewrite Algorithm 1 as follows

$$\mathbf{x}^{k+1} = \mathbf{W}_k(\mathbf{x}^k - \eta\mathbf{y}^k) \tag{5a}$$

$$\mathbf{y}^{k+1} = \mathbf{W}_k(\mathbf{y}^k + \alpha\nabla\mathbf{f}(\mathbf{x}^{k+1}) - \alpha\nabla\mathbf{f}(\mathbf{x}^k)). \tag{5b}$$

We now list the assumptions used in our analysis.

Assumption 2.2. The mixing matrix W is doubly stochastic and symmetric.

The Metropolis-Hastings algorithm (Hastings, 1970) can be used to construct mixing matrices from an undirected graph satisfying Assumption 2.2. Moreover, from Assumption 2.2, the mixing matrix W has a singular, maximum eigenvalue denoted as $\lambda_1 = 1$. All other eigenvalues are defined as $\{\lambda_i\}_{i=2}^n$. We define the mixing rate as $\lambda := \max_{i\in\{2,\dots,n\}}\{|\lambda_i|\}$.

Assumption 2.3. Each function $f_i : \mathbb{R}^m \to \mathbb{R}$ is L-smooth for $i \in \mathcal{V}$, i.e., $\|\nabla f_i(y) - \nabla f_i(z)\| \leq L\|y - z\|, \forall\, y, z \in \mathbb{R}^m$ for some $L > 0$. We assume there exists a $f^* \in \mathbb{R}$ such that $f(x) \geq f^*$.

We are now ready to state the main result of this paper on the convergence analysis of LU-GT.

Theorem 2.4 (Convergence of LU-GT)**.** *Let Assumptions 2.2 and 2.3 hold, and let, $T_o \in \mathbb{Z}_{\geq 0}$, $\eta > 0$, and $\alpha > 0$ with $\eta < O(1/T_o)$, and $\alpha < O((1-\lambda)/L)$ (Exact bounds found in (18), (19), (23), (25)). Then, for any $K \geq 1$, the output $\mathbf{x}^K$, of Algorithm 1 (LU-GT) with $\mathbf{x}^0 = (\mathbf{1} \otimes x^0)$ for any $x^0 \in \mathbb{R}^m$ has the following property:*

$$\frac{1}{K}\sum_{k=0}^{K-1}\left(\|\nabla f(\bar{x}^k)\|^2 + \|\overline{\nabla\mathbf{f}}(\mathbf{x}^k)\|^2\right) + \frac{L^2}{Kn}\sum_{k=0}^{K-1}\|\boldsymbol{\Phi}^k\|^2 \leq + \frac{8}{\eta\alpha K}\tilde{f}(\bar{x}^0) + \frac{3\alpha^2L^2T_o\zeta_0}{nK(1-\bar{\lambda})^2}, \tag{6}$$

Proof. The proof can be found in Appendix B. □

Note that the left-hand side of (7) has three main components. The first two indicate the asymptotic convergence to a stationary point, while the third term $\|\boldsymbol{\Phi}^k\|^2$ guarantees asymptotic consensus. If in Theorem 2.4, we consider a sufficiently well-connected graph where $1 \geq 2\sqrt{\lambda}$ and set $\alpha \propto (1-\lambda)/L$, and $\eta \propto 1/T_o$, then we obtain the convergence rate,

$$\frac{1}{K}\sum_{k=0}^{K-1}\left(\|\nabla f(\bar{x}^k)\|^2 + \|\overline{\nabla\mathbf{f}}(\mathbf{x}^k)\|^2\right) + \frac{L^2}{Kn}\sum_{k=0}^{K-1}\|\boldsymbol{\Phi}^k\|^2 \leq O\left(\frac{T_o\tilde{f}(\bar{x}^0)}{K} + \frac{T_o\zeta_0}{nK}\right). \tag{7}$$

The communication complexity of LU-GT is obtained by dividing the number of iterations K by T_o to find the number of communication rounds, i.e., $R = K/T_o$. Theorem 2.4 implies that LU-GT matches the same communication complexity ($R = O(1/\epsilon)$ for a desired accuracy $\epsilon > 0$) as (Karimireddy et al., 2020) for distributed (federated) setups. However, LU-GT allows arbitrary symmetric undirected network topologies (Assumptions 2.2).

3. Numerical Results

We simulate the performance of Algorithm 1 for the following least squares problem with a non-convex regularization term:

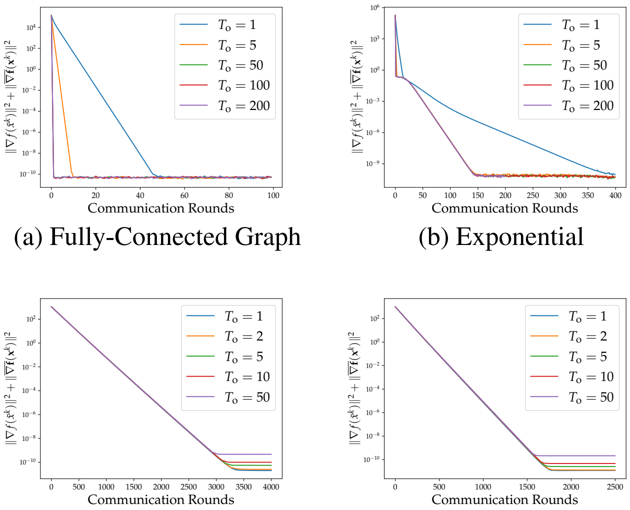
$$\min_x \frac{1}{n}\sum_{i=1}^n\|A_i x - b_i\|^2 + \rho\sum_{j=1}^m \frac{x(j)^2}{1 + x(j)^2}, \tag{8}$$

where $\{A_i, b_i\}$ is the local data held by agent i and $x(j)$ is the $j - th$ component of the parameter x. We consider two cases: 1) close to homogeneous, where local stationary points are different but sufficiently close; 2) heterogeneous, where no assumptions are made on the similarity of local stationary points. We generate $A_i \in \mathbb{R}^{p\times m}$ where $p = 500, m = 20$ with values drawn from $\mathcal{N}(0,1)$, a parameter vector $x_i^* \in \mathbb{R}^m$ with values drawn from $\mathcal{N}(0,1)$, and $b_i \in \mathbb{R}^p = A_i x_0^i + \gamma \times z_i$ where $z_i \in \mathbb{R}^p$ is drawn from $\mathcal{N}(0,1)$. This is a heterogeneous case. The difference for the close to homogeneous is that we draw A_i once such that $A_i = A_j, \forall i,j$. For the close to homogeneous case, we examine exponential and fully-connected graphs, while for the heterogeneous case, we examine star and ring graphs, all with 16 nodes. We set $\rho = 0.01, \gamma = 150$. Table 1 lists the manually optimized $\eta\alpha$ for each graph and T_{o} combination.

Table 1. Manually optimized $\eta\alpha$ used for each graph and T_{o} combination.

	$T_o = 1$	$T_o = 5$	$T_o = 50$	$T_o = 100$	$T_o = 200$
Complete	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}
Exponential	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}
	$T_o = 1$	$T_o = 2$	$T_o = 5$	$T_o = 10$	$T_o = 50$
Ring	2×10^{-5}	1×10^{-5}	0.4×10^{-5}	$.2 \times 10^{-5}$	0.04×10^{-5}
Star	$.4 \times 10^{-4}$	$.2 \times 10^{-4}$	$.08 \times 10^{-4}$	$.04 \times 10^{-4}$	$.008 \times 10^{-4}$

Our simulation results in Figure 1 reveal that for (sufficiently well-connected) graphs, LU-GT reduces communication costs up to a certain T_{o}. In addition, for the exponential graph, the benefits saturate much faster. For sparse networks, the hyperparameter tuning of $\eta\alpha$ matches the suggested inversely proportional relation with T_{o} predicted by the theory. In this scenario, communication costs are equivalent to no local updates, matching the analysis.



(a) Fully-Connected Graph

(b) Exponential

(c) Ring

(d) Star

Figure 1. Performance of LU-GT to solve (8) with varying T_{o}, $\alpha\eta$, and topologies.

4. Conclusions

We propose the algorithm LU-GT that incorporates local recursions into Gradient Tracking. Our analysis shows that LU-GT matches the same communication complexity as the Federated Learning setting but allows arbitrary network topologies. In addition, regardless of the number of local recursions, LU-GT incurs no additional bias term in the rate. We show reduced communication complexity in simulation for well-connected graphs. However, further refinement of the analysis is necessary to quantify the precise effect of local recursions on Gradient Tracking. It is still unclear under what regimes local updates reduce the communication cost and what the upper bound is on these local updates. Numerical results suggest that local updates might not benefit sparsely connected networks. Such explicit relations between network topologies and local updates are left for future work. While we focus on the non-convex setting in this work due to space constraints, we can extend our work to the convex setting. Another extension of the work we have done on LU-GT is accounting for the stochastic setting to determine if the analysis can reveal linear speedup similar to what has already been show for vanilla Gradient Tracking (Alghunaim & Yuan, 2022). Additionally, we can consider more sophisticated scenarios such as asynchronous updates (Assran et al., 2020) and varying the number of local updates throughout the progression of the algorithm. In the latter case, future studies may reveal scenarios in which performing many local updates initally and then increasing the communication frequency over iterations may improve the performance of LU-GT.

References

Alghunaim, S. A. and Yuan, K. A unified and refined convergence analysis for non-convex decentralized learning. *IEEE Transactions on Signal Processing*, 70:3264–3279, June 2022.

Assran, M., Aytekin, A., Feyzmahdavian, H., Johansson, M., and Rabbat, M. Advances in asynchronous parallel and distributed optimization, 2020.

Cattivelli, F. S. and Sayed, A. H. Diffusion LMS strategies for distributed estimation. *IEEE Trans. Signal Process*, 58(3):1035, 2010.

Chen, J. and Sayed, A. H. Distributed pareto optimization via diffusion strategies. *IEEE J. Sel. Topics Signal Process.*, 7(2):205–220, April 2013.

Di Lorenzo, P. and Scutari, G. Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, 2016.

Gorbunov, E., Hanzely, F., and Richtarik, P. Local SGD: Unified theory and new efficient methods. In Banerjee,

A. and Fukumizu, K. (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3556–3564. PMLR, 13–15 Apr 2021.

Hastings, W. K. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57 (1):97–109, 1970. ISSN 00063444.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. SCAFFOLD: Stochastic controlled averaging for federated learning. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5132–5143. PMLR, 13–18 Jul 2020.

Khaled, A., Mishchenko, K., and Richtárik, P. First analysis of local GD on heterogeneous data. *CoRR*, abs/1909.04715, 2019.

Khaled, A., Mishchenko, K., and Richtárik, P. Tighter theory for local SGD on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pp. 4519–4529. PMLR, 2020a.

Khaled, A., Mishchenko, K., and Richtarik, P. Tighter theory for local SGD on identical and heterogeneous data. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 4519–4529. PMLR, 26–28 Aug 2020b.

Koloskova, A., Loizou, N., Boreiri, S., Jaggi, M., and Stich, S. A unified theory of decentralized SGD with changing topology and local updates. In *International Conference on Machine Learning*, pp. 5381–5393, 2020.

Koloskova, A., Lin, T., and Stich, S. U. An improved analysis of gradient tracking for decentralized machine learning. *Advances in Neural Information Processing Systems*, 34:11422–11435, 2021.

Li, Z., Shi, W., and Yan, M. A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *IEEE Transactions on Signal Processing*, 67(17):4494–4506, Sept. 2019.

Lin, T., Stich, S. U., Patel, K. K., and Jaggi, M. Don't use large mini-batches, use local SGD. In *International Conference on Learning Representations*, 2020.

Lu, S. and Wu, C. W. Decentralized stochastic non-convex optimization over weakly connected time-varying digraphs. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5770–5774, 2020.

Mishchenko, K., Malinovsky, G., Stich, S., and Richtárik, P. Proxskip: Yes! local gradient steps provably lead to communication acceleration! finally! In *International Conference on Machine Learning*, 2022.

Mitra, A., Jaafar, R., Pappas, G. J., and Hassani, H. Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

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

Qu, G. and Li, N. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3):1245–1260, Sept. 2018.

Ram, S. S., Nedic, A., and Veeravalli, V. V. Distributed stochastic subgradient projection algorithms for convex optimization. *J. Optim. Theory Appl.*, 147(3):516–545, 2010.

Saadatniaki, F., Xin, R., and Khan, U. A. Decentralized optimization over time-varying directed graphs with row and column-stochastic matrices. *IEEE Transactions on Automatic Control*, 65(11):4769–4780, 2020.

Scutari, G. and Sun, Y. Distributed nonconvex constrained optimization over time-varying digraphs. *Mathematical Programming*, 176(1-2):497–544, 2019.

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

Stich, S. U. Local SGD converges fast and communicates little. In *International Conference on Learning Representations*, 2019.

Sun, Y., Scutari, G., and Daneshmand, A. Distributed optimization based on gradient tracking revisited: Enhancing convergence rate via surrogation. *SIAM Journal on Optimization*, 32(2):354–385, 2022.

Tang, H., Lian, X., Yan, M., Zhang, C., and Liu, J. D^2: Decentralized training over decentralized data. In *International Conference on Machine Learning*, pp. 4848–4856, Stockholm, Sweden, 2018.

Xu, J., Zhu, S., Soh, Y. C., and Xie, L. Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. In Proc. 54th *IEEE Conference on Decision and Control (CDC)*, pp. 2055–2060, Osaka, Japan, 2015.

Ying, B., Yuan, K., Hu, H., Chen, Y., and Yin, W. Bluefog: Make decentralized algorithms practical for optimization and deep learning. 2021.

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

Yuan, K., Ying, B., Zhao, X., and Sayed, A. H. Exact diffusion for distributed optimization and learning—part i: Algorithm development. *IEEE Transactions on Signal Processing*, 67(3):708–723, 2019. doi: 10.1109/TSP.2018.2875898.

Yuan, K., Alghunaim, S. A., Ying, B., and Sayed, A. H. On the influence of bias-correction on distributed stochastic optimization. *IEEE Transactions on Signal Processing*, 68:4352–4367, 2020.

Zhang, J., De Sa, C., Mitliagkas, I., and Ré, C. Parallel SGD: When does averaging help? *arXiv preprint arXiv:1606.07365*, 06 2016.

A. Transformation of Algorithm 1

We perform a series of transformations on (5) to simplify the analysis and accurately characterize the behavior of our algorithm. In particular, the trajectory of the average of agent parameters $\bar{x}^k$ is defined to show convergence of $\bar{x}^k$ to a first-order stationary point of (1). Motivated by (Alghunaim & Yuan, 2022), the deviation of agent parameters $\mathbf{x}^k$ from the average $\bar{\mathbf{x}}^k \triangleq \bar{x}^k \otimes \mathbf{1}_n$ and the deviation of the gradient tracking variable $\mathbf{y}^k$ from its average $\bar{\mathbf{y}}^k \triangleq \bar{y}^k \otimes \mathbf{1}_n$ are considered jointly as one augmented quantity, which simplifies the analysis.

Note that the mixing matrix W can be decomposed as

$$W = Q\Lambda Q^T = \begin{bmatrix} \frac{1}{\sqrt{n}}\mathbf{1} & \hat{Q} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \hat{\Lambda} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{n}}\mathbf{1}^T \\ \hat{Q}^T \end{bmatrix},$$

where $\hat{\Lambda} = \mathrm{diag}\{\lambda_i\}_{i=2}^n$, Q is a square orthogonal ($QQ^T = Q^TQ = I$), and $\hat{Q}$ is a matrix of size $n \times (n-1)$ such that $\hat{Q}\hat{Q}^T = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ and $\mathbf{1}^T\hat{Q} = 0$. From the above, we have

$$\mathbf{W} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T = \begin{bmatrix} \frac{1}{\sqrt{n}}\mathbf{1} \otimes I_m & \hat{\mathbf{Q}} \end{bmatrix} \begin{bmatrix} I_m & 0 \\ 0 & \hat{\boldsymbol{\Lambda}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{n}}\mathbf{1}^T \otimes I_m \\ \hat{\mathbf{Q}}^T \end{bmatrix},$$

where $\hat{\boldsymbol{\Lambda}} \triangleq \hat{\Lambda} \otimes I_m \in \mathbb{R}^{m(n-1)\times m(n-1)}$, $\mathbf{Q} \in \mathbb{R}^{mn\times mn}$ is orthogonal, and $\hat{\mathbf{Q}} \triangleq \hat{U} \otimes I_m \in \mathbb{R}^{mn\times m(n-1)}$ satisfies:

$$\hat{\mathbf{Q}}^T\hat{\mathbf{Q}}{=}\mathbf{I},\ \hat{\mathbf{Q}}\hat{\mathbf{Q}}^T{=}\mathbf{I}{-}\tfrac{1}{n}\mathbf{1}\mathbf{1}^T \otimes I_m,\ (\mathbf{1}^T \otimes I_m)\hat{\mathbf{Q}}{=}0. \tag{9}$$

Using (4), it follows that $\boldsymbol{\Lambda}_k \triangleq \boldsymbol{\Lambda}$ if $k \in \tau$ and $\boldsymbol{\Lambda}_k \triangleq \mathbf{I}$ otherwise. Equation (9) directly leads to

$$\begin{aligned} \|\hat{\mathbf{Q}}^T\mathbf{x}\|^2 &= \mathbf{x}^T\hat{\mathbf{Q}}\hat{\mathbf{Q}}^T\hat{\mathbf{Q}}\hat{\mathbf{Q}}^T\mathbf{x} = \|\hat{\mathbf{Q}}\hat{\mathbf{Q}}^T\mathbf{x}\|^2 = \|\mathbf{x} - \bar{\mathbf{x}}\|^2 \\ \|\hat{\mathbf{Q}}^T\mathbf{y}\|^2 &= \mathbf{y}^T\hat{\mathbf{Q}}\hat{\mathbf{Q}}^T\hat{\mathbf{Q}}\hat{\mathbf{Q}}^T\mathbf{y} = \|\hat{\mathbf{Q}}\hat{\mathbf{Q}}^T\mathbf{y}\|^2 = \|\mathbf{y} - \bar{\mathbf{y}}\|^2. \end{aligned}$$

In addition, we know that $\mathbf{Q} \triangleq \begin{bmatrix} \frac{1}{\sqrt{n}}\mathbf{1} \otimes I_m & \hat{\mathbf{Q}} \end{bmatrix}$. To recover the average $\bar{x}$ from the augmented vector $\mathbf{x}$, the following operation can be performed $(\frac{1}{n}\mathbf{1}^T \otimes I_m)\mathbf{x} = \bar{x}$. Hence, we multiply (5) by $\mathbf{Q}^T$ and simplify to get

$$\begin{aligned} \mathbf{Q}^T\mathbf{x}^{k+1} &= \boldsymbol{\Lambda}_k\mathbf{Q}^T(\mathbf{x}^k - \eta\mathbf{y}^k) \\ \mathbf{Q}^T\mathbf{y}^{k+1} &= \boldsymbol{\Lambda}_k\mathbf{Q}^T\mathbf{y}^k + \alpha\boldsymbol{\Lambda}_k\mathbf{Q}^T(\nabla\mathbf{f}(\mathbf{x}^{k+1}) - \nabla\mathbf{f}(\mathbf{x}^k)) \\ \begin{bmatrix} \bar{x}^{k+1} \\ \hat{\mathbf{Q}}^T\mathbf{x}^{k+1} \end{bmatrix} &= \boldsymbol{\Lambda}_k \left(\begin{bmatrix} \bar{x}^k \\ \hat{\mathbf{Q}}^T\mathbf{x}^k \end{bmatrix} - \begin{bmatrix} \eta\alpha\overline{\nabla\mathbf{f}}(\mathbf{x}^k) \\ \eta\hat{\mathbf{Q}}^T\mathbf{y}^k \end{bmatrix} \right) \\ \begin{bmatrix} \overline{\nabla\mathbf{f}}(\mathbf{x}^{k+1}) \\ \hat{\mathbf{Q}}^T\mathbf{y}^{k+1} \end{bmatrix} &{=}\boldsymbol{\Lambda}_k \begin{bmatrix} \overline{\nabla\mathbf{f}}(\mathbf{x}^k) \\ \hat{\mathbf{Q}}^T\mathbf{y}^k \end{bmatrix} + \alpha\boldsymbol{\Lambda}_k \begin{bmatrix} \overline{\nabla\mathbf{f}}(\mathbf{x}^{k+1}) - \overline{\nabla\mathbf{f}}(\mathbf{x}^k) \\ \hat{\mathbf{Q}}^T(\nabla\mathbf{f}(\mathbf{x}^{k+1}) - \nabla\mathbf{f}(\mathbf{x}^k)) \end{bmatrix}. \end{aligned}$$

Using the structure of $\boldsymbol{\Lambda}_k$, we then have

$$\bar{x}^{k+1}{=}\bar{x}^k - \eta\alpha\overline{\nabla\mathbf{f}}(\mathbf{x}^k) \tag{11a}$$

$$\hat{\mathbf{Q}}^T\mathbf{x}^{k+1}{=}\hat{\boldsymbol{\Lambda}}_k\hat{\mathbf{Q}}^T(\mathbf{x}^k - \eta\mathbf{y}^k) \tag{11b}$$

$$\hat{\mathbf{Q}}^T\mathbf{y}^{k+1}{=}\hat{\boldsymbol{\Lambda}}_k\hat{\mathbf{Q}}^T\mathbf{y}^k{+}\alpha\hat{\boldsymbol{\Lambda}}_k\hat{\mathbf{Q}}^T(\nabla\mathbf{f}(\mathbf{x}^{k+1}){-}\nabla\mathbf{f}(\mathbf{x}^k)). \tag{11c}$$

Observe that the average vector update (11a) has stepsizes α and η, the stepsize α comes from the fact that

$$\bar{y}^{k+1} = \bar{y}^k + \alpha(\overline{\nabla\mathbf{f}}(\mathbf{x}^{k+1}) - \overline{\nabla\mathbf{f}}(\mathbf{x}^k)) = \alpha\overline{\nabla\mathbf{f}}(\mathbf{x}^{k+1}), \tag{12}$$

where where $\bar{y}^k = \frac{1}{n}\sum_{i=1}^n y_i^k$ and the last step holds due to our initialization $\mathbf{y}^0 = \alpha\nabla\mathbf{f}(\mathbf{x}^0)$. Equation (11a) shows that the average of agent parameters, $\bar{x}$, is updated by performing a gradient descent step using the global gradient evaluated at the past average gradients. Then, $\bar{x}$ will converge to a stationary point in the limit. Therefore, if agents reach a consensus, this consensus will be a stationary point of (1). We then convert (11) into matrix notation

$$\bar{x}^{k+1} = \bar{x}^k - \eta\alpha\overline{\nabla\mathbf{f}}(\mathbf{x}^k) \tag{13a}$$

$$\begin{bmatrix} \hat{\mathbf{Q}}^T\mathbf{x}^{k+1} \\ \hat{\mathbf{Q}}^T\mathbf{y}^{k+1} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{\Lambda}}_k & -\eta\hat{\boldsymbol{\Lambda}}_k \\ \mathbf{0} & \hat{\boldsymbol{\Lambda}}_k \end{bmatrix} \begin{bmatrix} \hat{\mathbf{Q}}^T\mathbf{x}^k \\ \hat{\mathbf{Q}}^T\mathbf{y}^k \end{bmatrix} + +\alpha \begin{bmatrix} \mathbf{0} \\ \hat{\boldsymbol{\Lambda}}_k\hat{\mathbf{Q}}^T(\nabla\mathbf{f}(\mathbf{x}^{k+1}) - \nabla\mathbf{f}(\mathbf{x}^k)) \end{bmatrix}. \tag{13b}$$

By iterating (13) up to $r_k T_\text{o}$ where $r_k = \lfloor k/T_\text{o} \rfloor$ it follows that (5) can be rewritten as

$$\bar{x}^{k+1} = \bar{x}^k - \eta\alpha\overline{\nabla \mathbf{f}}(\mathbf{x}^k) \tag{14a}$$

$$\mathbf{\Phi}^{k+1} = \left(\prod_{l=k}^{r_k T_\text{o}} \mathbf{G}_l\right) \mathbf{\Phi}^{r_k T_\text{o}} + \alpha \mathbf{h}^{k+1} + \alpha \sum_{t=r_k T_\text{o}}^{k-1} \left(\prod_{l=k-1}^{t} \mathbf{G}_l\right) \mathbf{h}^{t+1}. \tag{14b}$$

where

$$\mathbf{\Phi}^k \triangleq \begin{bmatrix} \hat{\mathbf{Q}}^T \mathbf{x}^k \\ \hat{\mathbf{Q}}^T \mathbf{y}^k \end{bmatrix},$$

$$\mathbf{G}_k \triangleq \begin{bmatrix} \hat{\mathbf{\Lambda}}_k & -\eta\hat{\mathbf{\Lambda}}_k \\ \mathbf{0} & \hat{\mathbf{\Lambda}}_k \end{bmatrix},$$

$$\mathbf{h}^{k+1} \triangleq \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{\Lambda}}_k \hat{\mathbf{Q}}^T (\nabla \mathbf{f}(\mathbf{x}^{k+1}) - \nabla \mathbf{f}(\mathbf{x}^k)) \end{bmatrix}.$$

In the next Section, we analyze and bound the trajectory of the augmented consensus quantity $\|\mathbf{\Phi}^k\|$ necessary to establish the convergence of Algorithm 1.

Remark A.1. The product notation $\prod_{l=k-1}^{t} \mathbf{G}_l$ start from the large indices to the smaller, which is different from sum notation where order is not important. As a simple, example, consider a recursion of the form $s^{k+1} = A_k s^k$. Then, we have for example $s^4 = A_3 s^3 = A_3 A_2 s^2 = A_3 A_2 A_1 s^1$.

B. Analysis on Convergence of Algorithm 1

In this section we prove our main result in Theorem 2.4. We start by introducing a series of technical lemmas that will help us build the desired result. Lemma B.1 and Lemma B.2 quantify the effect of local steps and the mixing matrix W on the augmented consensus quantity. Lemma B.3 provides a bound of the deviation of the parameter $\mathbf{x}^k$ between iterations. This is needed in Lemma B.4 to bound the quantity $\mathbf{h}^k$ used in the bound on the augmented consensus quantity.

Lemma B.1. *For iterates t, and k of Algorithm 1 where $r_k T_\text{o} < t < k-1$ and $k-1, t \notin \tau$, the following matrix inequality holds*

$$\left\| \prod_{l=k-1}^{t} \mathbf{G}_l \right\| \leq 1 + \eta(k-1-t) < 1 + \eta T_\text{o}. \tag{15}$$

Proof.

$$\left\| \prod_{l=k-1}^{t} \mathbf{G}_l \right\| = \left\| \begin{bmatrix} \mathbf{I} & -\eta(k-1-t)\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \right\|$$
$$= \left\| \mathbf{I} + \begin{bmatrix} \mathbf{0} & -\eta(k-1-t)\mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right\| \leq 1 + \eta(k-1-t) \leq 1 + \eta T_\text{o}.$$

The first equality follows from multiplying $\mathbf{G}_l$ from $l = t$ to $l = k-1$. The second equality follows from directly decomposing the result matrix product as a sum. The final step uses the sub-additive property of matrix norms. □

Lemma B.2. *Suppose that Assumption 2.2 holds. For an iterate k of Algorithm 1 where $r_k T_\text{o} < k$ and $k \notin \tau$, the following matrix inequality holds*

$$\left\| \prod_{l=k}^{r_k T_\text{o}} \mathbf{G}_l \right\| \leq \lambda(1 + \eta T_\text{o}). \tag{16}$$

Proof.

$$\prod_{l=k}^{r_k T_\text{o}} \mathbf{G}_l \triangleq \mathbf{C} = \begin{bmatrix} \hat{\mathbf{\Lambda}} & -\eta\hat{\mathbf{\Lambda}} \\ \mathbf{0} & \hat{\mathbf{\Lambda}} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\eta(k - r_k T_\text{o})\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$
$$= \begin{bmatrix} \hat{\mathbf{\Lambda}} & -\eta(k - r_k T_\text{o})\hat{\mathbf{\Lambda}} - \eta\hat{\mathbf{\Lambda}} \\ \mathbf{0} & \hat{\mathbf{\Lambda}} \end{bmatrix}.$$

This result directly follows from multiplying $\mathbf{G}_l$ from $l = r_k T_\text{o}$ to $l = k$.

There exists a coordinate of transformation matrix $\mathbf{R}$ such that $\mathbf{R}^T\mathbf{C}\mathbf{R} = \text{blkdiag}\{\mathbf{C}\}_{i=2}^n$, where

$$\begin{aligned}\mathbf{C}_i &= \begin{bmatrix}\lambda_i & -\eta(k - r_kT_\text{o})\lambda_i - \eta\lambda_i \\ 0 & \lambda_i\end{bmatrix} \\ &= \lambda_i\left(I + \begin{bmatrix}0 & -\eta(k - r_kT_\text{o}) - \eta \\ 0 & 0\end{bmatrix}\right).\end{aligned}$$

To get the above result, we factored out λ_i and decomposed the matrix as a sum of matrices. Hence,

$$\begin{aligned}\|\mathbf{C}_i\| &\le \lambda_i(1 + \eta(k - r_kT_\text{o}) + \eta) \\ \|\mathbf{C}\| &\le \lambda(1 + \eta(k - r_kT_\text{o}) + \eta) \le \lambda(1 + \eta(T_\text{o})).\end{aligned}$$

Here we first used the sub-additive property of matrix norms. Then, we took advantage of the block-diagonal structure of $\mathbf{R}^T\mathbf{C}\mathbf{R}$ and the fact that $\|\mathbf{R}\| = 1$. □

Lemma B.3. *Let $0 < \eta, \alpha < 1$ and $k \ge 0$. Then, an iterate $\mathbf{x}^k$ of Algorithm 1 has the following property:*

$$\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 = \begin{cases}4\|\mathbf{\Phi}^k\|^2 + 4n\eta^2\alpha^2\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2 & k \in \tau, \\ 4\eta^2\|\mathbf{\Phi}^k\|^2 + 4n\eta^2\alpha^2\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2 & \textit{else}.\end{cases}$$

where n is the number of agents.

Proof. Depending on k, we have two possibilities

$$\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 = \begin{cases}\|(\mathbf{W} - \mathbf{I})\mathbf{x}^k - \eta\mathbf{W}\mathbf{y}^k\|^2 & \text{when } k \in \tau, \\ \|\eta\mathbf{y}^k\|^2 & \text{otherwise.}\end{cases}$$

We start by bounding the first case:

$$\begin{aligned}\|(\mathbf{W} - \mathbf{I})\mathbf{x}^k - \eta\mathbf{W}\mathbf{y}^k\|^2 &= \|(\mathbf{W} - \mathbf{I})(\mathbf{x}^k - (\mathbf{1} \otimes \bar{x}^k)) - \eta\mathbf{W}\mathbf{y}^k\|^2 \\ &\le 4\|\mathbf{x}^k - \bar{\mathbf{x}}^k\|^2 + 2\|\eta\mathbf{y}^k\|^2 \\ &\le 4\|\mathbf{x}^k - \bar{\mathbf{x}}^k\|^2 + 4\eta^2\|\mathbf{y}^k - \bar{\mathbf{y}}^k\|^2 + 4\eta^2\|\bar{\mathbf{y}}^k\|^2\end{aligned}$$

The first equality adds and subtracts $\mathbf{1} \otimes \bar{x}^k$ inside the norm. We take advantage of the fact that $\mathbf{W}(\mathbf{1} \otimes \bar{x}^k)) = \mathbf{1} \otimes \bar{x}^k$. In the first inequality, we use $\|a + b\|^2 \le 2\|a\|^2 + 2\|b\|^2$ twice and then use Assumption 2.2 to upper bound the spectral norm of $\mathbf{W}$ by 1. In the final inequality, we use $\|a + b\|^2 \le 2\|a\|^2 + 2\|b\|^2$. Using (12), we have

$$\|(\mathbf{W} - \mathbf{I})\mathbf{x}^k - \mathbf{W}\mathbf{y}^k\|^2 \le 4\|\mathbf{x}^k - \bar{\mathbf{x}}^k\|^2 + 4\eta^2\|\mathbf{y}^k - \bar{\mathbf{y}}^k\|^2 + 4n\eta^2\alpha^2\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2.$$

Using the properties in (9) the following upper bound on the consensus error holds

$$\|\mathbf{\Phi}^k\|^2 = \left\|\begin{bmatrix}\hat{\mathbf{Q}}^T\mathbf{x}^k \\ \hat{\mathbf{Q}}^T\mathbf{y}^k\end{bmatrix}\right\|^2 = \|\mathbf{x}^k - \bar{\mathbf{x}}^k\|^2 + \|\mathbf{y}^k - \bar{\mathbf{y}}^k\|^2.$$

Since $0 < \eta < 1$ it follows that

$$\|\mathbf{x}^k - \bar{\mathbf{x}}^k\|^2 + \eta^2\|\mathbf{y}^k - \bar{\mathbf{y}}^k\|^2 \le \|\mathbf{\Phi}^k\|^2. \tag{17}$$

Hence,

$$\|(\mathbf{W} - \mathbf{I})\mathbf{x}^k - \mathbf{W}\mathbf{y}^k\|^2 \le 4\|\mathbf{\Phi}^k\|^2 + 4n\eta^2\alpha^2\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2.$$

We now bound the second case

$$\begin{aligned}\|\eta\mathbf{y}^k\|^2 &= \eta^2\|\mathbf{y}^k - \bar{\mathbf{y}}^k + \bar{\mathbf{y}}^k\|^2 \\ &\le 2\eta^2\|\hat{\mathbf{Q}}^T\mathbf{y}^k\|^2 + 2n\eta^2\alpha^2\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2 \\ &\le 4\eta^2\|\mathbf{\Phi}^k\|^2 + 4n\eta^2\alpha^2\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2.\end{aligned}$$

In the first inequality, we apply $\|a + b\|^2 \le 2\|a\|^2 + 2\|b\|^2$ and use (9) on the term $\|\bar{\mathbf{y}}^k - \bar{\mathbf{y}}^k\|$. Then, we use (12) on the term $\bar{\mathbf{y}}^k$. In the final inequality, we use (17). □

Lemma B.4. *Let Assumptions 2.2 and 2.3 hold. For an iteration $k \notin \tau$, step-size $\alpha > 0$, smoothness parameter L defined in Assumption 2.3, constant $1 > \eta > 0$, and number of local iterations T_{o}, the following inequality holds*

$$\begin{aligned}\|\mathbf{h}^{k+1}\|^2 + \left\|\sum_{t=r_kT_{\mathrm{o}}}^{k-1}\left(\prod_{l=k-1}^{t}\mathbf{G}_l\right)\mathbf{h}^{t+1}\right\|^2 &\leq 8L^2\eta^2T_{\mathrm{o}}(1+\eta T_{\mathrm{o}})^2\sum_{t=r_kT_{\mathrm{o}}+1}^{k}(\|\mathbf{\Phi}^t\|^2+n\alpha^2\|\overline{\nabla\mathbf{f}}(\mathbf{x}^t)\|^2)\\ &\quad + 8L^2\lambda^2(1+\eta T_{\mathrm{o}})(\|\mathbf{\Phi}^{r_kT_{\mathrm{o}}}\|^2+n\eta^2\alpha^2\|\overline{\nabla\mathbf{f}}(\mathbf{x}^{r_kT_{\mathrm{o}}})\|^2).\end{aligned}$$

Proof. First, we use $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ to obtain,

$$\begin{aligned}\|\mathbf{h}^{k+1}\|^2 + \left\|\sum_{t=r_kT_{\mathrm{o}}}^{k-1}\left(\prod_{l=k-1}^{t}\mathbf{G}_l\right)\mathbf{h}^{t+1}\right\|^2 &\leq \|\mathbf{h}^{k+1}\|^2 + 2\left\|\left(\prod_{l=k-1}^{r_kT_{\mathrm{o}}}\mathbf{G}_l\right)\mathbf{h}^{r_kT_{\mathrm{o}}+1}\right\|^2 + 2\left\|\sum_{t=r_kT_{\mathrm{o}}+1}^{k-1}\left(\prod_{l=k-1}^{t}\mathbf{G}_l\right)\mathbf{h}^{t+1}\right\|^2\\ &\leq L^2\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2+2L^2\lambda^2(1+\eta T_{\mathrm{o}})\|\mathbf{x}^{r_kT_{\mathrm{o}}+1}-\mathbf{x}^{r_kT_{\mathrm{o}}}\|^2\\ &\quad + 2L^2T_{\mathrm{o}}(1+\eta T_{\mathrm{o}})^2\sum_{t=r_kT_{\mathrm{o}}+1}^{k-1}\left\|\mathbf{x}^{t+1}-\mathbf{x}^t\right\|^2\\ &\leq 4L^2\eta^2(\|\mathbf{\Phi}^k\|^2+n\alpha^2\|\overline{\nabla\mathbf{f}}(\mathbf{x}^k)\|^2)\\ &\quad + 8L^2\lambda^2(1+\eta T_{\mathrm{o}})(\|\mathbf{\Phi}^{r_kT_{\mathrm{o}}}\|^2+n\eta^2\alpha^2\|\overline{\nabla\mathbf{f}}(\mathbf{x}^{r_kT_{\mathrm{o}}})\|^2)\\ &\quad + 8L^2\eta^2T_{\mathrm{o}}(1+\eta T_{\mathrm{o}})^2\sum_{t=r_kT_{\mathrm{o}}+1}^{k-1}(\|\mathbf{\Phi}^t\|^2+n\alpha^2\|\overline{\nabla\mathbf{f}}(\mathbf{x}^t)\|^2)\\ &\leq 8L^2\lambda^2(1+\eta T_{\mathrm{o}})(\|\mathbf{\Phi}^{r_kT_{\mathrm{o}}}\|^2+n\eta^2\alpha^2\|\overline{\nabla\mathbf{f}}(\mathbf{x}^{r_kT_{\mathrm{o}}})\|^2)\\ &\quad + 8L^2\eta^2T_{\mathrm{o}}(1+\eta T_{\mathrm{o}})^2\sum_{t=r_kT_{\mathrm{o}}+1}^{k}(\|\mathbf{\Phi}^t\|^2+n\alpha^2\|\overline{\nabla\mathbf{f}}(\mathbf{x}^t)\|^2).\end{aligned}$$

In the second inequality, we used Lemma B.1, Lemma B.2, and Assumption 2.3. In the third inequality, we used Lemma B.3. In the fourth inequality, we group similar terms. □

Next, we find a bound on the consensus inequality to later use in the descent inequality. Note that we define $\sum_{t=r_kT_{\mathrm{o}}}^{k}(\cdot)$ as zero if $r_kT_{\mathrm{o}} > k-1$.

Lemma B.5 (Consensus Inequality)**.** *Let Assumptions 2.2 and 2.3 hold and*

$$\eta < \min\left\{1, \frac{(1-\sqrt{\lambda})}{(\sqrt{\lambda})(T_{\mathrm{o}})}\right\}, \tag{18}$$

$$\alpha \leq \min\left\{\sqrt{\frac{(1-\lambda)(1-\theta)}{16L^2\lambda}}, \sqrt{\frac{(\bar{\lambda}-\bar{\lambda}^2)(1-\theta)}{8L^2\eta^2T_{\mathrm{o}}^2}}\right\}, \tag{19}$$

hold. Define $\theta = \lambda(1+\eta T_{\mathrm{o}})^2 < 1$. Then, the output of Algorithm (1) satisfies the following inequality

$$\frac{1}{K}\sum_{k=0}^{K-1}\|\mathbf{\Phi}^k\|^2 \leq \frac{(1-\bar{\lambda})(\frac{1}{K}\sum_{k=0}^{K-1}\bar{\lambda}^{r_k})}{1-\bar{\lambda}-e_1T_{\mathrm{o}}}\|\mathbf{\Phi}^0\|^2 + \left(\frac{e_2T_{\mathrm{o}}}{K(1-\bar{\lambda}-e_1T_{\mathrm{o}})}\right)\sum_{k=0}^{K-1}\left(\|\overline{\nabla\mathbf{f}}(\mathbf{x}^k)\|^2 + \|\nabla f(\bar{x}^k)\|^2\right), \tag{20}$$

where $\|\mathbf{\Phi}^k\|^2 = \|\mathbf{x}^k - \bar{\mathbf{x}}^k\|^2 + \|\mathbf{y}^k - \bar{\mathbf{y}}^k\|^2$, $r_k \triangleq \lfloor k/T_{\mathrm{o}}\rfloor$, $e_1 \triangleq \frac{8L^2\eta^2\alpha^2T_{\mathrm{o}}(1+\eta T_{\mathrm{o}})^2}{(1-\theta)}$, and $e_2 \triangleq \frac{8nL^2\eta^2\alpha^4T_{\mathrm{o}}(1+\eta T_{\mathrm{o}})^2}{(1-\theta)}$.

Proof of Lemma B.5. We take the norm of (14) and apply Jensen's inequality for any $0 < \theta < 1$.

$$
\begin{aligned}
\|\mathbf{\Phi}^{k+1}\|^2 \le{}& \frac{1}{\theta}\left\|\left(\prod_{l=k}^{r_kT_\mathrm{o}} \mathbf{G}_l\right)\mathbf{\Phi}^{r_kT_\mathrm{o}}\right\|^2 + \frac{2\alpha^2}{(1-\theta)}\left(\|\mathbf{h}^{k+1}\|^2 + \left\|\sum_{t=r_kT_\mathrm{o}}^{k-1}\left(\prod_{l=k-1}^{t}\mathbf{G}_l\right)\mathbf{h}^{t+1}\right\|^2\right)\\
\le{}& \frac{\lambda^2(1+\eta T_\mathrm{o})^2}{\theta}\left\|\mathbf{\Phi}^{r_kT_\mathrm{o}}\right\|^2 + \frac{8L^2\alpha^2(1+\eta T_\mathrm{o})^2}{(1-\theta)}\left(\sum_{t=r_kT_\mathrm{o}+1}^{k} T_\mathrm{o}\eta^2\|\mathbf{\Phi}^t\|^2 + \lambda^2\|\mathbf{\Phi}^{r_kT_\mathrm{o}}\|^2\right)\\
&+\frac{8nL^2\eta^2\alpha^4(1+\eta T_\mathrm{o})^2}{(1-\theta)}\left(\sum_{t=r_kT_\mathrm{o}+1}^{k} T_\mathrm{o}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2 + \lambda^2\|\overline{\nabla \mathbf{f}}(\mathbf{x}^{r_kT_\mathrm{o}})\|^2\right).
\end{aligned}
$$

In the second inequality, we applied the results from Lemma B.2 and Lemma B.4. Set

$$
\theta = \lambda(1+\eta T_\mathrm{o})^2 < 1 \Rightarrow \eta < \frac{1-\sqrt{\lambda}}{\sqrt{\lambda}(T_\mathrm{o})}.
$$

Moreover, define $e_1 \triangleq \frac{8L^2\eta^2\alpha^2 T_\mathrm{o}(1+\eta T_\mathrm{o})^2}{(1-\theta)}$, and $e_2 \triangleq \frac{8nL^2\eta^2\alpha^4 T_\mathrm{o}(\eta T_\mathrm{o})^2}{(1-\theta)}$, then

$$
\|\mathbf{\Phi}^{k+1}\|^2 \le (\lambda + \frac{\lambda^2 e_1}{T_\mathrm{o}\eta^2})\left\|\mathbf{\Phi}^{r_kT_\mathrm{o}}\right\|^2 + e_1\sum_{t=r_kT_\mathrm{o}+1}^{k}\|\mathbf{\Phi}^t\|^2 + e_2\left(\sum_{t=r_kT_\mathrm{o}+1}^{k}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^t)\|^2 + \frac{\lambda^2}{T_\mathrm{o}}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^{r_tT_\mathrm{o}})\|^2\right).
$$

Choose α such that

$$
\lambda + \frac{\lambda^2 e_1}{T_\mathrm{o}\eta^2} \le \frac{1+\lambda}{2} \Rightarrow \alpha \le \sqrt{\frac{(1-\lambda)(1-\theta)}{16L^2\lambda}}.
$$

Defining $\bar\lambda = (1+\lambda)/2$ and observing that $\frac{\lambda^2}{T_\mathrm{o}} < 1$, we have

$$
\|\mathbf{\Phi}^{k+1}\|^2 \le \bar\lambda\|\mathbf{\Phi}^{r_kT_\mathrm{o}}\|^2 + e_1\sum_{t=r_kT_\mathrm{o}+1}^{k}\|\mathbf{\Phi}^t\|^2 + e_2\sum_{t=r_kT_\mathrm{o}}^{k}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^t)\|^2. \tag{21}
$$

When $k = r_kT_\mathrm{o} - 1$, we have

$$
\|\mathbf{\Phi}^{r_kT_\mathrm{o}}\|^2 \le \bar\lambda\|\mathbf{\Phi}^{(r_k-1)T_\mathrm{o}}\|^2 + e_1\sum_{t=(r_k-1)T_\mathrm{o}}^{r_kT_\mathrm{o}-1}\|\mathbf{\Phi}^t\|^2 + e_2\sum_{t=(r_k-1)T_\mathrm{o}}^{r_kT_\mathrm{o}-1}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^t)\|^2.
$$

Substitute the above into (21) and iterate to find

$$
\begin{aligned}
\|\mathbf{\Phi}^{k+1}\|^2 \le{}& \bar\lambda^{r_k+1}\|\mathbf{\Phi}^0\|^2 + e_1\left(\sum_{t=r_kT_\mathrm{o}}^{k}\|\mathbf{\Phi}^t\|^2 + \bar\lambda\sum_{t=(r_k-1)T_\mathrm{o}}^{r_kT_\mathrm{o}-1}\|\mathbf{\Phi}^t\|^2 + \cdots + \bar\lambda^{r_k}\sum_{t=0}^{T_\mathrm{o}-1}\|\mathbf{\Phi}^t\|^2\right)\\
&+ e_2\left(\sum_{t=r_kT_\mathrm{o}}^{k}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^t)\|^2 + \bar\lambda\sum_{t=(r_k-1)T_\mathrm{o}}^{r_kT_\mathrm{o}-1}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^t)\|^2 + \cdots + \bar\lambda^{r_k}\sum_{t=0}^{T_\mathrm{o}-1}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2\right).
\end{aligned}
$$

Recall that $r_k = \lfloor \frac{k}{T_\mathrm{o}} \rfloor$. Thus, we introduce the notation

$$
\bar\lambda^{(k,t)} \triangleq \begin{cases} 0 & t \le -1\\ 1 & r_kT_\mathrm{o} \le t \le k\\ \bar\lambda & (r_k-1)T_\mathrm{o} \le t \le r_kT_\mathrm{o}-1\\ \vdots & \vdots\\ \bar\lambda^{r_k} & 0 \le t \le T_\mathrm{o}-1. \end{cases}
$$

We can then describe the previous bound more compactly as

$$\|\mathbf{\Phi}^k\|^2 \leq \bar{\lambda}^{r_k+1}\|\mathbf{\Phi}^0\|^2 + e_1\sum_{t=0}^{k-1}\bar{\lambda}^{(k-1,t)}\|\mathbf{\Phi}^t\|^2 + e_2\sum_{t=0}^{k-1}\bar{\lambda}^{(k-1,t)}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^t)\|^2.$$

when setting $k+1$ as k. Then, we average over $k=0,...,K-1$ and upper bound the result as follows

$$\begin{aligned}\frac{1}{K}\sum_{k=0}^{K-1}\|\mathbf{\Phi}^k\|^2 &\leq \frac{1}{K}\sum_{k=0}^{K-1}\bar{\lambda}^{r_k+1}\|\mathbf{\Phi}^0\|^2 + \frac{e_1}{K}\sum_{k=0}^{K-1}\sum_{t=0}^{k-1}\bar{\lambda}^{(k-1,t)}\|\mathbf{\Phi}^t\|^2 + \frac{e_2}{K}\sum_{k=0}^{K-1}\sum_{t=0}^{k-1}\bar{\lambda}^{(k-1,t)}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^t)\|^2 \\ &= \frac{1}{K}\sum_{k=0}^{K-1}\bar{\lambda}^{r_k+1}\|\mathbf{\Phi}^0\|^2 + \frac{e_1}{K}\sum_{t=0}^{K-1}\sum_{k=t}^{K-1}\bar{\lambda}^{(k-1,t)}\|\mathbf{\Phi}^t\|^2 + \frac{e_2}{K}\sum_{t=0}^{K-1}\sum_{k=t}^{K-1}\bar{\lambda}^{(k-1,t)}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^t)\|^2 \\ &= \frac{1}{K}\sum_{k=0}^{K-1}\bar{\lambda}^{r_k+1}\|\mathbf{\Phi}^0\|^2 + \frac{e_1}{K}\sum_{t=0}^{K-1}\|\mathbf{\Phi}^t\|^2\sum_{k=t}^{K-1}\bar{\lambda}^{(k-1,t)} + \frac{e_2}{K}\sum_{t=0}^{K-1}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^t)\|^2\sum_{k=t}^{K-1}\bar{\lambda}^{(k-1,t)} \\ &\leq \frac{1}{K}\sum_{k=0}^{K-1}\bar{\lambda}^{r_k+1}\|\mathbf{\Phi}^0\|^2 + \frac{e_1T_{\mathrm{o}}}{K(1-\bar{\lambda})}\sum_{k=0}^{K-1}\|\mathbf{\Phi}^k\|^2 + \frac{e_2T_{\mathrm{o}}}{K(1-\bar{\lambda})}\sum_{k=0}^{K-1}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2.\end{aligned}$$

In the first equality, we rearrange the order of the summation. In the second equality, we rearrange the terms in the summation based on their index. In the second inequality, we upper bound $\sum_{k=0}^{K-1}\sum_{k=t}^{K-1}\bar{\lambda}^{(k-1,t)}$ with $T_{\mathrm{o}}/(1-\bar{\lambda})$ and change the indexing from t to k afterwards. Therefore,

$$\left(1-\frac{e_1T_{\mathrm{o}}}{1-\bar{\lambda}}\right)\frac{1}{K}\sum_{k=0}^{K-1}\|\mathbf{\Phi}^k\|^2 \leq \frac{1}{K}\sum_{k=0}^{K-1}\bar{\lambda}^{r_k+1}\|\mathbf{\Phi}^0\|^2 + \left(\frac{e_2T_{\mathrm{o}}}{K(1-\bar{\lambda})}\right)\sum_{k=0}^{K-1}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2,$$

and with further simplification, we have

$$\left(\frac{1-\bar{\lambda}-e_1T_{\mathrm{o}}}{1-\bar{\lambda}}\right)\frac{1}{K}\sum_{k=0}^{K-1}\|\mathbf{\Phi}^k\|^2 \leq \frac{1}{K}\sum_{k=0}^{K-1}\bar{\lambda}^{r_k+1}\|\mathbf{\Phi}^0\|^2 + \left(\frac{e_2T_{\mathrm{o}}}{K(1-\bar{\lambda})}\right)\sum_{k=0}^{K-1}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2.$$

By imposing the following assumption on α

$$\frac{1-\bar{\lambda}-e_1T_{\mathrm{o}}}{1-\bar{\lambda}} \geq 1-\bar{\lambda} \Rightarrow \alpha \leq \sqrt{\frac{(\bar{\lambda}-\bar{\lambda}^2)(1-\theta)}{8L^2\eta^2T_{\mathrm{o}}^2}}, \tag{22}$$

it follows that

$$\begin{aligned}\frac{1}{K}\sum_{k=0}^{K-1}\|\mathbf{\Phi}^k\|^2 &\leq \frac{(1-\bar{\lambda})(\frac{1}{K}\sum_{k=0}^{K-1}\bar{\lambda}^{r_k+1})}{1-\bar{\lambda}-e_1T_{\mathrm{o}}}\|\mathbf{\Phi}^0\|^2 + \left(\frac{e_2T_{\mathrm{o}}}{K(1-\bar{\lambda}-e_1T_{\mathrm{o}})}\right)\sum_{k=0}^{K-1}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2 \\ &\leq \frac{(1-\bar{\lambda})(\frac{1}{K}\sum_{k=0}^{K-1}\bar{\lambda}^{r_k+1})}{1-\bar{\lambda}-e_1T_{\mathrm{o}}}\|\mathbf{\Phi}^0\|^2 + \left(\frac{e_2T_{\mathrm{o}}}{K(1-\bar{\lambda}-e_1T_{\mathrm{o}})}\right)\sum_{k=0}^{K-1}\left(\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2 + \|\nabla f(\bar{x}^k)\|^2\right).\end{aligned}$$

□

We are now ready to state the proof of Theorem 2.4.

Proof of Theorem 2.4. Following similar arguments as in Lemma 3 of (Alghunaim & Yuan, 2022) and imposing

$$\alpha \leq \frac{1}{2L}, \tag{23}$$

we have the following inequality

$$f(\bar{x}^{k+1}) \leq f(\bar{x}^k) - \frac{\eta\alpha}{2}\|\nabla f(\bar{x}^k)\|^2 - \frac{\eta\alpha}{4}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2 + \frac{\eta\alpha L^2}{2n}\|\mathbf{\Phi}^k\|^2.$$

Reorganize and lower bound the left-hand side to find

$$\frac{\eta\alpha}{4}\big(\|\nabla f(\bar{x}^k)\|^2+\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2\big)\le f(\bar{x}^k)-f(\bar{x}^{k+1})+\frac{\eta\alpha L^2\|\boldsymbol{\Phi}^k\|^2}{2n}.$$

□

Next, subtract and add f^* and set $\tilde{f}(\bar{x}^k) = f(\bar{x}^k) - f^*$, then

$$\|\nabla f(\bar{x}^k)\|^2+\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2\le\frac{4}{\eta\alpha}\big(f(\bar{x}^k)-f(\bar{x}^{k+1})\big)+\frac{2L^2}{n}\|\boldsymbol{\Phi}^k\|^2.$$

Sum both sides from $k = 0, ..., K-1$ and divide by K

$$\frac{1}{K}\sum_{k=0}^{K-1}\Big(\|\nabla f(\bar{x}^k)\|^2 + \|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2\Big)\le \frac{4}{\eta\alpha K}\sum_{k=0}^{K-1}(\tilde{f}(\bar{x}^k) - \tilde{f}(\bar{x}^{k+1})) + \frac{2L^2}{nK}\sum_{k=0}^{K-1}\|\boldsymbol{\Phi}^k\|^2. \tag{24}$$

Multiplying (20) from Lemma B.5 by c, a constant to be defined later, and adding it to the above equation, we then have the following

$$\begin{aligned}\frac{1}{K}\sum_{k=0}^{K-1}\Big(\|\nabla f(\bar{x}^k)\|^2 + \|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2\Big)+c\frac{1}{K}\sum_{k=0}^{K-1}\|\boldsymbol{\Phi}^k\|^2 \le{}& \frac{4}{\eta\alpha K}\sum_{k=0}^{K-1}(\tilde{f}(\bar{x}^k) - \tilde{f}(\bar{x}^{k+1})) + \frac{2L^2}{nK}\sum_{k=0}^{K-1}\|\boldsymbol{\Phi}^k\|^2\\ &+ c\frac{(1-\bar{\lambda})(\frac{1}{K}\sum_{k=0}^{K-1}\bar{\lambda}^{r_k})}{1-\bar{\lambda}-e_1T_{\mathrm{o}}}\|\boldsymbol{\Phi}^0\|^2\\ &+ c\left(\frac{e_2T_{\mathrm{o}}}{K(1-\bar{\lambda}-e_1T_{\mathrm{o}})}\right)\sum_{k=0}^{K-1}\left(\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2 + \|\nabla f(\bar{x}^k)\|^2\right).\end{aligned}$$

Rearranging and setting $c = \frac{3L^2}{n}$ we find

$$\begin{aligned}\left(1-\frac{3L^2e_2T_{\mathrm{o}}}{n(1-\bar{\lambda}-e_1T_{\mathrm{o}})}\right)\frac{1}{K}\sum_{k=0}^{K-1}\Big(\|\nabla f(\bar{x}^k)\|^2 + \|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2\Big)+\left(\frac{L^2}{n}\right)\frac{1}{K}\sum_{k=0}^{K-1}\|\boldsymbol{\Phi}^k\|^2&\\ \le \frac{4}{\eta\alpha K}\sum_{k=0}^{K-1}(\tilde{f}(\bar{x}^k) - \tilde{f}(\bar{x}^{k+1})) + c\frac{(1-\bar{\lambda})(\frac{1}{K}\sum_{k=0}^{K-1}\bar{\lambda}^{r_k})}{1-\bar{\lambda}-e_1T_{\mathrm{o}}}\|\boldsymbol{\Phi}^0\|^2.&\end{aligned}$$

Require

$$\frac{1}{2}\le\left(1-\frac{3L^2e_2T_{\mathrm{o}}}{n(1-\bar{\lambda}-e_1T_{\mathrm{o}})}\right)\Rightarrow\alpha\le\sqrt[4]{\frac{(1-\bar{\lambda})^2(1-\theta)}{48L^4\eta^2T_{\mathrm{o}}^2}}. \tag{25}$$

Then, we have

$$\frac{1}{K}\sum_{k=0}^{K-1}\left(\|\nabla f(\bar{x}^k)\|^2 + \|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2\right)+\frac{L^2}{Kn}\sum_{k=0}^{K-1}\|\boldsymbol{\Phi}^k\|^2\le\frac{8}{\eta\alpha K}\tilde{f}(\bar{x}^0)+\frac{6L^2(1-\bar{\lambda})(\sum_{k=0}^{K-1}\bar{\lambda}^{r_k})}{nK(1-\bar{\lambda}-e_1T_{\mathrm{o}})}\|\boldsymbol{\Phi}^0\|^2.$$

Assume that the initialization for $x_1, x_2, ..., x_n$ is identical. Then $\mathbf{x}^0 = \mathbf{1}\otimes x^0$ (for some $x^0\in\mathbb{R}^d$). As a result, $\mathbf{x}^0 = \bar{\mathbf{x}}^0$ meaning $\|\hat{\mathbf{Q}}^T\mathbf{x}^0\|^2 = 0$. Then,

$$\|\boldsymbol{\Phi}^0\|^2 = \|\hat{\mathbf{Q}}^T\mathbf{y}^0\|^2 = \left\|\alpha\hat{\mathbf{Q}}^T\nabla\mathbf{f}(\mathbf{x}^0)\right\|^2 = \alpha^2\|\nabla\mathbf{f}(\bar{\mathbf{x}}^0) - \mathbf{1}\otimes\overline{\nabla \mathbf{f}}(\bar{\mathbf{x}}^0)\|^2.$$

Define $\zeta_0 = \|\nabla\mathbf{f}(\bar{\mathbf{x}}^0) - \mathbf{1}\otimes\overline{\nabla \mathbf{f}}(\bar{\mathbf{x}}^0)\|^2$. We also upper bound $\sum_{k=0}^{K-1}\bar{\lambda}^{r_k}$ with $T_{\mathrm{o}}/(1-\bar{\lambda})$, a repeating geometric sequence and use (22). Then, the desired relation follows.