An Improved Analysis of Gradient Tracking for Decentralized Machine Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study stochastic decentralized machine learning optimization, where training data is distributed over n workers that communicate only locally with a small number of neighbors. Our focus is the setting where worker's local data are heterogeneous. This is an important setting motivated by federated learning applications and is challenging in practice for many decentralized algorithms.
Gradient Tracking (GT) is a popular decentralized optimization algorithm that is agnostic to data dissimilarity in both convex and non-convex problems. While the existing theoretical results show the independence of the data heterogeneity level, the convergence rates are not optimal in regards other problem parameters that are of importance in practice. We provide a tighter analysis of the GT method in the stochastic setting with a novel proof technique. We are the first to prove that in the non-convex setting with stochastic noise σ, GT asymptotically converges with an optimal $\mathcal{O}\left(\sigma/\sqrt{nT}\right)$ rate, where T denotes the number of iterations.

1 Introduction

Methods that train machine learning models on decentralized data offer many advantages over traditional centralized approaches in core aspects such as data ownership, privacy, fault tolerance and scalability [10]. Many current efforts in this direction are spearheaded under the banner of federated learning [15, 25, 24, 10], where a central entity orchestrates the training and collects updates from the participating devices. Fully decentralized methods, that do not rely on a central coordinator and that communicate only with neighbors in an arbitrary communication topology, are still in their infancy.

With the work of Lian et al. [20] on decentralized SGD (D-SGD), research on decentralized training methods for machine learning models intensified. To tackle practical challenges, methods for time-varying topologies [31, 2, 14] and methods with communication compression [38, 44, 13, 41] have been designed. One of the most challenging aspect when training over decentralized data is data-heterogeneity, i.e. training data that is in a non-IID fashion distributed over the devices (for instance in data-center training) or generated on non-IID fashion on client devices [17, 11, 18, 19]. In particular, the D-SGD method is provably affected by the heterogenity [14]. In contrast, the D^2 method by Tang et al. [40] does provably not depend on data-heterogenity and can be applied when the communication topology remains fixed and does not change over time.

Decentralized optimization methods have been studied for decades in the optimization and control community [42, 28, 45, 5] and it might appear puzzling why there is a need to design new methods to tackle ML problems, instead of just resorting to classical methods. For instance, in contributions

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

preceding [39], Lorenzo and Scutari [22] and Nedić et al. [26] developed methods to address the data-heterogeneity (we denote these methods as *gradient tracking* (GT) methods henceforth).

While it is well understood that GT methods do not depend on the data heterogeneity, and for instance converge linearly on distributed strongly convex problem instances without noise [22, 26]. However, when applying these methods in the machine learning settings, we need to understand how they are impacted by noise and how they behave on non-convex tasks. Moreover, we need precise estimates of the convergence rates of GT, in order to select the best training algorithm for a particular problem.

In this paper, we develop a new, and improved, analysis of the gradient tracking algorithm (improving over all known results in both the convex and non-convex cases) with a novel proof technique (which could be of independent interest). Alongside with the parallel contribution [48] that developed a tighter analysis of the D^2 algorithm, we now have a tighter understanding in which setting GT works well and in which ones it does not, and can compare to D-SGD and D^2.

In this paper we improve over all existing results that analyze the GT algorithm. Specifically, we prove a weaker dependence on the graph spectral gap. We give comparison of GT convergence rates in the Tables 1 and 2.

Contributions. Our main contributions can be summarized as:

- We prove better complexity estimates for the GT algorithm than known before with a new proof technique (which might be of independent interest).
- We show that in the presence of stochastic noise, the leading term in the convergence rate of GT is optimal—we are the first to derive this in the non-convex setting—and matching the unimprovable rate of all-reduce mini-batch SGD.
- In the non-asymptotic regime (of importance in practice), the convergence rate depends on the network topology. By defining new graph parameters, we can give a tighter description of this dependency, explaining why the worst case behavior is rarely observed in practice. We verify this dependence in numerical experiments.
- We prove that the convergence rate of GT matches the best known rate for D^2 [48] on all topologies where D^2 [40] can be applied.[1]

2 Related Work

Decentralized Optimization. Decentralized optimization has a vast literature and it tracks back at least to [42]. The large number of decentralized optimization methods [28, 9] are based on gossip averaging [12, 46, 3]. These methods are usually work well in non-convex setup and are used for deep learning training [2], [20], [39]. There exists other methods, such as based on alternating direction method of multipliers (ADMM) [45, 8], dual averaging [5, 29, 35], primal-dual methods [1], or block-coordinate methods for generalized linear models [7].

Decentralized Optimization with Heterogeneous Functions. There exists several algorithms that are agnostic to data-heterogeneity. Notably, EXTRA [36], decentralized primal-dual [1], convergence behaviors do not depend on the data heterogeneity and achieve linear convergence in the strongly convex noiseless setting. However, these algorithms are not designed to be used for non-convex tasks.

D^2 [40, 48] (also known as exact diffusion [49, 50]) and Gradient Tracking (GT) [22] (also known as NEXT [22] or DIGing [26]) are the algorithms that both are agnostic to data heterogeneity level, work with the stochastic noise, and for the non-convex functions, the setting of utmost importance in machine learning due to deep learning applications. The limitations of D^2 algorithm is that it is not shown to work on changing topologies, moreover only for restricted fixed topologies with negative eigenvalue of a mixing matrix being bounded from below by $-\frac{1}{3}$. Other authors proposed algorithms that perform well on heterogeneous DL tasks [21, 52], but without provable guarantees for heterogeneity.

[1] In D^2 the smallest eigenvalue of the mixing matrix W must bounded from below: $\min_i \lambda_i(W) \geq -\frac{1}{3}$.

Gradient Tracking. There is a vast literature on Gradient Tracking method itself. A tracking mechanism was used by Zhu and Martínez [54] as a way to track the average of a distributed continuous process. Lorenzo and Scutari [22] applied this technique to track the gradients, and analyzed its asymptotic behavior in the non-convex setting with a time-varying topologies. Nedić et al. [26] analyze GT (named as DIGing) in the strongly convex noiseless case with a time-varying network. Qu and Li [34] extend the GT analysis to the non-convex, weakly-convex and strongly convex case without stochastic noise. Nedić et al. [27] allow the different stepsizes on different workers. Yuan et al. [51] analyze asymptotic behavior of GT for dynamic optimization. Pu and Nedić [33] studied the GT method on stochastic problems and strongly convex objectives. Further, Xin et al. [47] analyze asymptotic behavior of GT with stochastic noise. For non-convex stochastic functions GT was analyzed by Zhang and You [53] and Lu et al. [23]. Li et al. [16] combine GT with variance reduction to achieve linear convergence in the stochastic case. Tziotis et al. [43] obtain second order guarantees for GT.

3 Setup

We consider optimization problems where the objective function is distributed across n nodes

$$\min \left[f(\mathbf{x}) := \sum_{i=1}^{n} \left[f_i(\mathbf{x}) = \mathbb{E}_{\xi \sim \mathcal{D}_i} F_i(\mathbf{x}, \xi) \right] \right], \qquad \mathbf{x} \in \mathbb{R}^d, \tag{1}$$

where f_i is the local functions available to the node i, $i \in \{1, \dots n\}$. Each f_i is a stochastic function $f_i(\mathbf{x}) = \mathbb{E}_{\xi \sim \mathcal{D}_i} F_i(\mathbf{x}, \xi)$ with access only to stochastic gradients $\nabla F_i(\mathbf{x}, \xi)$. This setting covers empirical risk minimization problems with $\mathcal{D}_i$ being a uniform distribution over the local training dataset. It also covers deterministic optimization when $F_i(\mathbf{x}, \xi) = f_i(\mathbf{x}) \ \forall \xi$.

We consider optimization over a decentralized network, i.e. when there is an underlying communication graph $G = (V, E)$, $|V| = n$, each of the nodes (i.e. device) can communicate only along the edges E. In decentralized optimization it is convenient to parameterize communication by a mixing matrix $W \in \mathbb{R}^{n \times n}$, where $w_{ij} = 0$ if and only if nodes i and j are not communicating, $(i, j) \notin E$.

Definition 1 (Mixing Matrix)**.** *A matrix with non-negative entries $W \in [0, 1]^{n \times n}$ that is symmetric ($W = W^\top$) and doubly stochastic ($W\mathbf{1} = \mathbf{1}$, $\mathbf{1}^\top W = \mathbf{1}^\top$).*

3.1 Notation

We use the notation $\mathbf{x}_i^{(t)} \in \mathbb{R}^d, \mathbf{y}_i^{(t)} \in \mathbb{R}^d$ to denote the iterates and the tracking sequence, respectively, on node i at time step t. For vectors $\mathbf{z}_i \in \mathbb{R}^d$ ($\mathbf{z}_i$ could for instance be $\mathbf{x}_i^{(t)}$ or $\mathbf{y}_i^{(t)}$) defined for $i \in [n]$ we denote by $\bar{\mathbf{z}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{z}_i$.

We use both vector and matrix notation whenever it is more convenient. For vectors $\mathbf{z}_i \in \mathbb{R}^d$ defined for $i \in [n]$ we denote by a capital letter the matrix with columns $\mathbf{z}_i$, formally

$$Z := [\mathbf{z}_1, \dots, \mathbf{z}_n] \in \mathbb{R}^{d \times n}, \qquad \bar{Z} := [\bar{\mathbf{z}}, \dots, \bar{\mathbf{z}}] \equiv Z \tfrac{1}{n} \mathbf{1}\mathbf{1}^\top, \qquad \Delta Z = Z - \bar{Z}. \tag{2}$$

We extend this definition to gradients of (1), with $\nabla F(X^{(t)}, \xi^{(t)}), \nabla f(X^{(t)}) \in \mathbb{R}^{d \times n}$:

$$\nabla F(X^{(t)}, \xi^{(t)}) = \left[\nabla F_1(\mathbf{x}_1^{(t)}, \xi_1^{(t)}), \dots, \nabla F_n(\mathbf{x}_n^{(t)}, \xi_n^{(t)}) \right]$$
$$\nabla f(X^{(t)}) = \left[\nabla f(\mathbf{x}_1^{(t)}), \dots, \nabla f(\mathbf{x}_n^{(t)}) \right].$$

3.2 Algorithm

The Gradient Tracking algorithm (or NEXT, DIGing) can be re-written as

$$\begin{pmatrix} X^{(t+1)} \\ \gamma Y^{(t+1)} \end{pmatrix} = \begin{pmatrix} W & -W \\ 0 & W \end{pmatrix} \begin{pmatrix} X^{(t)} \\ \gamma Y^{(t)} \end{pmatrix} + \gamma \begin{pmatrix} 0 \\ \nabla F(X^{(t+1)}, \xi^{(t+1)}) - \nabla F(X^{(t)}, \xi^{(t)}) \end{pmatrix} \tag{GT}$$

in matrix notation. Here and $X^{(t)} \in \mathbb{R}^{d \times n}$ denotes the iterates, $Y^{(t)} \in \mathbb{R}^{d \times n}$, with $Y^{(0)} = \nabla F(X^{(t)}, \xi^{(t)})$ the sequence of tracking variables, and $\gamma > 0$ is a stepsize. This update is summarized in Algorithm 1.

Algorithm 1 GRADIENT TRACKING

input : Initial values $\mathbf{x}_i^{(0)} \in \mathbb{R}^d$ on each node $i \in [n]$, communication graph $G = ([n], E)$ and mixing matrix W, stepsize γ, initialize $\mathbf{y}_i^{(0)} = \nabla f(\mathbf{x}_i^{(0)})$, $\mathbf{g}_i^{(0)} = \mathbf{y}_i^{(0)}$, $\forall i$

1: **for** t **in** $0 \dots T-1$ **do** {*in parallel for all workers* $i \in [n]$}
2: send $\mathbf{x}_i^{(t)}$, $\mathbf{y}_i^{(t)}$ to the neighbours of node i
3: $\mathbf{x}_i^{(t+1)} = \sum_{j:\{i,j\}\in E} w_{ij} \left(\mathbf{x}_j^{(t)} - \eta_t \mathbf{y}_j^{(t)}\right)$
4: Sample $\xi_i^{(t+1)}$, compute gradient $\mathbf{g}_i^{(t+1)} = \nabla F_i(\mathbf{x}_i^{(t+1)}, \xi_i^{(t+1)})$
5: $\mathbf{y}_i^{(t+1)} = \sum_{j:\{i,j\}\in E} w_{ij} \mathbf{y}_j^{(t)} + \mathbf{g}_i^{(t+1)} - \mathbf{g}_i^{(t)}$
6: **end for**

Each node i stores and updates two variables $\mathbf{x}_i^{(t)}$ and $\mathbf{y}_i^{(t)}$. Variable $\mathbf{x}_i^{(t)}$ represents current model parameters and is updated on line 3 with a decentralized SGD update but using $\mathbf{y}_i^{(t)}$ instead of a gradient. Variable $\mathbf{y}_i^{(t)}$ tracks the average of all local gradients on line 5. Intuitively, the algorithm is agnostic to the functions heterogeneity because $\mathbf{y}_i^{(t)}$ is 'close' to the full gradient of f (suppose we would replace line 5 with exact averaging, then $\mathbf{y}_t^{(t+1)} = \frac{1}{n}\mathbf{g}_i^{(t+1)}$. For further discussion of the tracking mechanism refer to [22, 26, 33].

3.3 Assumptions

In our analysis we use the following standard assumptions.

Assumption 1 (Mixing Matrix)**.** *Let $\lambda_i(W)$, $i \in [n]$ denote the eigenvalues of W with $1 = \lambda_1(W) > \lambda_2(W) \geq \dots \geq \lambda_n(W) > -1$ and define the spectral gap $\delta = 1 - \max\{|\lambda_2(W)|, |\lambda_n(W)|\}$. We also define*

$$p = 1 - \max\{|\lambda_2(W)|, |\lambda_n(W)|\}^2 \,, \qquad\qquad c = 1 - |\lambda_n(W)|^2 \,. \tag{3}$$

We assume that $p > 0$ (and consequently $c > 0$).

This assumption ensures that the consensus distance decreases linearly after each averaging step, i.e. $\left\|XW - \bar{X}\right\|_F^2 \leq (1-p)\left\|X - \bar{X}\right\|_F^2$, $\forall X \in \mathbb{R}^{d\times n}$. We also can conclude that $c \geq p$ for all mixing matrices W and they are equal only when $|\lambda_n(W)| \geq |\lambda_2(W)|$. Assuming a lower bound on p (or equivalently δ) is a standard assumption in the literature.

Assumption 2 (L-smoothness)**.** *Each function $f_i(\mathbf{x})\colon \mathbb{R}^d \to \mathbb{R}$, $i \in [n]$ is differentiable and there exists a constant $L \geq 0$ such that for each $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$:*

$$\|\nabla f_i(\mathbf{y}) - \nabla f_i(\mathbf{x})\| \leq L \|\mathbf{x} - \mathbf{y}\| \,. \tag{4}$$

Some of the results are for the (strongly) convex functions.

Assumption 3 (μ-strong convexity)**.** *Each function $f_i(\mathbf{x})\colon \mathbb{R}^d \to \mathbb{R}$, $i \in [n]$ is μ-strongly convex for constant $\mu \geq 0$, i.e. for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$:*

$$f_i(\mathbf{x}) - f_i(\mathbf{y}) + \tfrac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 \leq \langle \nabla f_i(\mathbf{x}), \mathbf{x} - \mathbf{y}\rangle \,. \tag{5}$$

Assumption 4 (Bounded noise)**.** *We assume that there exists constant σ s.t. $\forall \mathbf{x}_1, \dots \mathbf{x}_n \in \mathbb{R}^d$*

$$\tfrac{1}{n}\sum_{i=1}^n \mathbb{E}_{\xi_i} \left\|\nabla F_i(\mathbf{x}_i, \xi_i) - \nabla f_i(\mathbf{x}_i)\right\|_2^2 \leq \sigma^2 \,. \tag{6}$$

We discuss possible relaxations of these assumptions in Section 4.1 below.

4 Convergence results

We now present our novel convergence results for GT in Section 4.1, and . We also give a proof sketch to explain the key difficulties and technical novelty compared to prior results in Section 6 below.

4.1 Main theorem—GT Convergence in the general case

Theorem 1. *For GT algorithm 1 with a mixing matrix as in Definition 1, under Assumptions 1, 2, 4, after T iterations, if $T > \frac{2}{p}\log\left(\frac{50}{p}(1+\log\frac{1}{p})\right)$, there exists a constant stepsize $\gamma_t = \gamma$ such that the error is bounded as*

Non-convex:

$$\frac{1}{T+1}\sum_{t=0}^{T}\left\|\nabla f(\bar{\mathbf{x}}^{(t)})\right\|_2^2 \leq \tilde{\mathcal{O}}\left(\sqrt{\frac{LF_0\hat{\sigma}^2}{nT}} + \left(\frac{\hat{\sigma}LF_0}{\sqrt{p}cT}\right)^{2/3} + \frac{LF_0\sqrt{P+1}}{pcT}\right),$$

Strongly-convex: *Under additional Assumption 3 with $\mu > 0$, it holds*

$$\sum_{t=0}^{T}\frac{w_t}{W_T}\left[\mathbb{E}\, f(\bar{\mathbf{x}}^{(t)}) - f^\star\right] \leq \tilde{\mathcal{O}}\left(\frac{\bar{\sigma}^2}{\mu nT} + \frac{L\bar{\sigma}^2}{\mu^2 pc^2T^2} + \frac{LR_0}{pc}\exp\left[-\frac{\mu pcT}{L}\right]\right), \tag{7}$$

Weakly-convex: *Under Assumptions 3 with $\mu \geq 0$, it holds*

$$\frac{1}{T+1}\sum_{t=0}^{T}\left[\mathbb{E}\, f(\bar{\mathbf{x}}^{(t)}) - f^\star\right] \leq \tilde{\mathcal{O}}\left(\sqrt{\frac{R_0^2\bar{\sigma}^2}{nT}} + \left(\frac{\bar{\sigma}\sqrt{L}R_0^2}{\sqrt{p}cT}\right)^{2/3} + \frac{LR_0^2}{pcT}\right), \tag{8}$$

where $F_0 = f(\bar{\mathbf{x}}^{(0)}) - f^\star$, $R_0 = \left\|\mathbf{x}^{(0)} - \mathbf{x}^\star\right\|$.

From these results we see that the leading term in the convergence rate (assuming $\bar{\sigma} > 0$) is not affected by the graph parameters. Moreover, in this term we see a linear speedup in n, the number of workers. The leading terms of all three results match with the convergence estimates for all-reduce mini-batch SGD [4, 37]. The higher order terms depend on p and c and we observe that the middle term is more strongly impacted by c than the last term (c^2p vs. cp). We will discuss the dependency on the graph parameters c, p more carefully below in Section 6.1 and Section 7.

Possible Relaxations of the Assumptions. Before moving on to the proofs, we mention briefly a few possible relaxations of the assumptions that are possible with only slight adaptions of the proof. These extensions can be addressed with known techniques and are omitted for conciseness. We give here the necessary references for completeness.

- **Bounded Gradient Assumption I.** The uniform bound on the stochastic noise in Assumption 4 could be relaxed by allowing the noise to grow with the gradient norm [14, Assumption 3b].
- **Bounded Gradient Assumption II.** In the convex setting it has been observed that σ^2 can be replaced with $\sigma_\star^2 := \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\xi_i}\left\|\nabla F_i(\mathbf{x}^\star,\xi_i) - \nabla f_i(\mathbf{x}^\star)\right\|_2^2$, the noise at the optimum. However, this requires smoothness of each $F_i(\mathbf{x},\xi)$, $\xi \in \mathcal{D}_i$, which is stronger than our Assumption 2. For the technique see [32].
- **Time-varying Graphs and Local Updates.** Several works studied decentralized methods with either arbitrary time-varying topologies (random, or following a determined sequence) or local updates (intermittent communication). These can both be captured in the model presented in [14, Assumption 4] or [30, Assumption 1] and could be added to our proof as well.
- **Different mixing for X and Y.** In Algorithm 1 both the $\mathbf{x}$ and $\mathbf{y}$ iterates are averaged on the same communication topology (the same mixing matrix). This can be relaxed by allowing for two separate matrices. This follows from inspecting our proof below.

4.2 Faster convergence on consensus functions

We now state an additional result, which improves Theorem 2 on the consensus problem, defined as

$$\min\left[f(\mathbf{x}) = \sum_{i=1}^{n}\left[f_i(\mathbf{x}) := \tfrac{1}{2}\left\|\mathbf{x} - \boldsymbol{\mu}_i\right\|^2\right]\right], \tag{9}$$

for vectors $\boldsymbol{\mu}_i \in \mathbb{R}^d$, $i \in [n]$ and optimal solution $\mathbf{x}^\star = \frac{1}{n}\sum_{i=1}^n \boldsymbol{\mu}_i$. Note that there is no stochastic noise and contrast to Theorem 2, the convergence rate in this case does not depend on c.

Theorem 2. *Let f be as in* (9) *let Assumption 1 hold. Then there exists a stepsize $\gamma \leq p$ such that it holds $\frac{1}{n}\sum_{i=1}^n \left\|\mathbf{x}_i^{(T)} - \mathbf{x}^\star\right\|^2 \leq \epsilon$, for the iterates GT 1 and any $\epsilon > 0$, after at most $T = \tilde{\mathcal{O}}\left(p \log \frac{1}{\epsilon}\right)$ iterations.*

5 Proof sketch of the main theorem

Here we give a proof sketch for Theorem 1, for the special case of strongly convex objectives. We give all proof details in the Appendix and highlight the main technical difficulties and novel techniques.

Key Lemma. It is very common—and useful— to write the iterates in the form $X^{(t)} = \bar{X}^{(t)} + (X^{(t)} - \bar{X}^{(t)})$, where $\bar{X}^{(t)}$ denotes the matrix with the average over the nodes. We can then separately analyze $\bar{X}^{(t)}$ and the consensus difference $\Delta X^{(t)} := (X^{(t)} - \bar{X}^{(t)})$ (and $\Delta Y^{(t)} := (Y^{(t)} - \bar{Y}^{(t)})$). From the update equation (GT) we see that

$$\begin{pmatrix} \Delta X^{(t+1)} \\ \gamma \Delta Y^{(t+1)} \end{pmatrix} = \underbrace{\begin{pmatrix} \tilde{W} & -\tilde{W} \\ 0 & \tilde{W} \end{pmatrix}}_{J} \underbrace{\begin{pmatrix} \Delta X^{(t)} \\ \gamma \Delta Y^{(t)} \end{pmatrix}}_{\Phi_t} + \gamma \underbrace{\begin{pmatrix} 0 \\ (I - \frac{\mathbf{1}\mathbf{1}^\top}{n}) \left(\nabla F(X^{t+1}, \xi^{t+1}) - \nabla F(X^t, \xi^t)\right) \end{pmatrix}}_{E_t},$$

in short

$$\Psi_{t+1} = J\Psi_t + \gamma E_t \,. \tag{10}$$

We could immediately adapt the proof technique from [14] if it would hold that the spectral radius of J is smaller than one. However, this is not the case, in general $\|J\| > 1$.

We now observe that for any integer $i \geq 0$:

$$J^i = \begin{pmatrix} \tilde{W}^i & -i\tilde{W}^i \\ 0 & \tilde{W}^i \end{pmatrix} \qquad \left\|J^i\right\|^2 = \left\|\tilde{W}^i\right\|^2 + i^2 \left\|\tilde{W}^i\right\|^2 \leq (1-p)^i + i^2(1-p)^i \,, \tag{11}$$

with Assumption 1. We can now formulate a key lemma:

Lemma 3 (Contraction)**.** *For any integer $\tau \geq \frac{2}{p}\log\left(\frac{50}{p}(1 + \log\frac{1}{p})\right)$ it holds that $\|J^\tau\|^2 \leq \frac{1}{2}$.*

While the constants in this lemma are chosen to easy the presentation, most important for us is that after $\tau = \tilde{\Theta}\left(\frac{1}{p}\right)$ communication rounds, old parameter values (from τ steps ago) get averaged by a constant factor. We can alternatively write the statement of Lemma 3 as

$$\left\|J^\tau Z - \bar{Z}\right\|_F^2 \leq \frac{1}{2}\left\|Z - \bar{Z}\right\|_F^2 \,, \forall Z \in \mathbb{R}^{2d \times n} \,.$$

This resembles [14, Assumption 4] and the proof now follows the same pattern. A few crucial differences remain, as the result in [14] depends on a data-dissimilarity parameter, which we can avoid by carefully estimating the tracking errors. For completeness, we sketch the outline and give all details in the appendix.

Average Sequence. First, we consider the average sequences $\bar{X}^{(t)}$ and $\bar{Y}^{(t)}$. As all columns of these matrices are equal, we can equivalently consider a single column only: $\bar{\mathbf{x}}^{(t)}$ and $\bar{\mathbf{y}}^{(t)}$.

Lemma 4 (Average)**.** *It holds that*

$$\bar{\mathbf{y}}^{(t)} = \frac{1}{n}\sum_{i=1}^n \nabla F_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)}) \,, \qquad \bar{\mathbf{x}}^{(t+1)} = \bar{\mathbf{x}}^{(t)} - \gamma \frac{1}{n}\sum_{i=1}^n \nabla F_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)}) \,. \tag{12}$$

This follows directly from the update (GT) and the fact that $\bar{X} = \mathbf{W}\bar{X}$ for doubly stochastic mixing matrices.

The update of $\bar{\mathbf{x}}^{(t)}$ in (12) looks almost identical to one step of mini-batch SGD (on a complete graph). The average sequence behaves almost as a SGD sequence:

Lemma 5 (Descent lemma, [14, Lemma 8])**.** *Under the Assumptions of Theorem 1, the averages $\bar{\mathbf{x}}^{(t)} := \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^{(t)}$ of the iterates of Algorithm 1 with the stepsize $\gamma \leq \frac{1}{12L}$ satisfy*

$$\mathbb{E}_{\boldsymbol{\xi}} \left\|\bar{\mathbf{x}}^{(t+1)} - \mathbf{x}^\star\right\|^2 \leq \left(1 - \frac{\gamma\mu}{2}\right)\left\|\bar{\mathbf{x}}^{(t)} - \mathbf{x}^\star\right\|^2 + \frac{\gamma^2\sigma^2}{n} - \gamma e_t + \gamma\frac{3L}{n}\sum_{i=1}^n \left\|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t)}\right\|^2 , \tag{13}$$

where $e_t = f(\bar{\mathbf{x}}^{(t)}) - f^\star$ and $\mathbb{E}_{\boldsymbol{\xi}}$ denotes the expectation over $\boldsymbol{\xi}_1^{(t)}, \ldots, \boldsymbol{\xi}_n^{(t)}$.

Consensus Distance. The main difficulty comes from estimating the consensus distance $\|\Psi_t\|^2$, in the notation introduced in (10). Note that

$$\|\Psi_t\|^2 = \frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{x}_i^{(t)} - \bar{\mathbf{x}}^{(t)}\right\|_2^2 + \frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{y}_i^{(t)} - \bar{\mathbf{y}}^{(t)}\right\|_2^2 .$$

By unrolling (10) for τ steps,

$$\Phi_{t+\tau} = J^{\tau}\Phi_t + \gamma\sum_{j=1}^{\tau-1} J^{\tau-j}E_{t+j-1} .$$

By taking the Frobenius norm, and carefully estimating the norm of the error term $\left\|\sum_{j=1}^{\tau-1} J^{\tau-j}E_{t+j-1}\right\|_F^2$, and Lemma 3 we can derive a recursion for the consensus distance.

Lemma 6 (Consensus distance recursion)**.** *There are exists absolute constants $C_1, C_2 > 0$ such that*

$$\mathbb{E}\,\|\Phi_{t+\tau}\|_F^2 \leq \frac{1}{2}\,\mathbb{E}\,\|\Phi_t\|_F^2 + \gamma^2\frac{C_1\tau}{c^2}\sum_{j=0}^{\tau}\mathbb{E}\left\|\nabla f(X^{t+j}) - \nabla f(X^{\star})\right\|_F^2 + \gamma^2\frac{C_2\tau}{c^2}\sigma^2 . \tag{14}$$

It remains to unroll (13) and (14). □

Proof sketch of Theorem 2. Using the matrix notation introduced above, the iterations of GT on problem (9) can be written in a simple form:

$$\begin{pmatrix}\Delta X^{(t+1)}\\ \gamma\Delta Y^{(t+1)}\end{pmatrix} = \underbrace{\begin{pmatrix}\tilde{W} & -\tilde{W}\\ \gamma\,(W-I) & (1-\gamma)\tilde{W}\end{pmatrix}}_{J'}\begin{pmatrix}\Delta X^{(t)}\\ \gamma\Delta Y^{(t)}\end{pmatrix} .$$

Similar as above, also the matrix J' is not a contraction operator, but in contrast to J it is diagonalizable: $J' = Q\Lambda Q^{-1}$ for some Q and diagonal Λ. It follows that $\|(J')^t\|^2 = \left\|Q\Lambda^t Q^{-1}\right\|^2$ is decreasing as $(1-p)^t\,\|Q\|^2\,\left\|Q^{-1}\right\|^2$. With this observation, the proof simplifies.

6 Discussion

We now provide a discussion of these results.

6.1 Parameter c

The convergence rate in Theorem 1 depends on the parameter c, that in the worst case could be as small as p. In this case our theoretical result does not improve over existing results for the strongly convex case. However, for many graphs in practice parameter c is bounded by a constant.

Remark 7 (Lower bound on c.)**.** *Let W be a mixing matrix with diagonal entries (self-weights) $w_{ii} \geq \delta > 0$. Then $\lambda_n(W) \geq 2\delta - 1$ and $c \geq 4\delta$.*

This follows from Gershgorin's circle theorem [6] and we give the proof in the appendix. For many choices of W considered in practice, most notably when graph G has constant node-degree and the weights w_{ij} are chosen by a popular Metropolis-Hastings rule, i.e. $w_{ij} = w_{ji} = \min\left\{\frac{1}{\deg(i)+1}, \frac{1}{\deg(j)+1}\right\}$ for $(i,j) \in E$, $w_{ii} = 1 - \sum_{j=1}^{n} w_{ij} \geq \frac{1}{\max_{j\in[n]}\deg(j)}$ see also [46, 3].

6.2 Comparison to prior GT literature.

Tables 1 and 2 compare our theoretical convergence rates in strongly convex and non convex settings. Our result tightens all existing prior work.

6.3 Comparison to other methods.

We now compare our complexity estimate of GT to D-SGD and D^2 in the strongly convex case. Analogous observations hold for the other cases too.

Comparison to D-SGD. A popular algorithm for decentralized optimization is D-SGD [20] that converges as [14]:

Table 1: Important advances for Gradient Tracking in the strongly convex case.

Reference	rate of convergence to ϵ-accuracy	considered stochastic noise
Nedić et al. [26]	$\mathcal{O}\left(\exp\left[-\frac{\mu^3 p^2 T}{L^3}\right]\right)$	✗
Qu and Li [34]	$\mathcal{O}\left(\exp\left[-\frac{\mu^2 p^2 T}{L^2}\right]\right)$	✗
Pu and Nedić [33]	$\tilde{\mathcal{O}}\left(\frac{\sigma^2}{\mu^2 nT} + \frac{L^2\sigma^2}{\mu^2 p^3 T^2} + \frac{\mathcal{O}_{\sigma,L,\mu,p}(1)}{T^2}\right)^a$	✓
this work	$\tilde{\mathcal{O}}\left(\frac{\sigma^2}{\mu nT} + \frac{L\sigma^2}{\mu^2 pc^2 T^2} + \frac{LR_0}{pc}\exp\left[-\frac{\mu pcT}{L}\right]\right)$	✓

[a] $\mathcal{O}_{\sigma,L,\mu,p}(1)$ is a constant that is independent of T, but can depend on other parameters, such as σ, μ, L, p

Table 2: Important advances for Gradient Tracking in the non-convex case.

Reference	rate of convergence to ϵ-accuracy	considered stochastic noise
Lorenzo and Scutari [22]	asymptotic convergence guarantees	✗
Zhang and You [53]	$\mathcal{O}\left(\frac{\sigma}{\sqrt{T}} + \frac{1}{p^3 T}\right)$	✓
Lu et al. [23]	$\mathcal{O}\left(\frac{C_1 + C_2\sigma}{\sqrt{T}}\right)^a$	✓
Qu and Li [34]	$\mathcal{O}\left(\frac{L}{p^2 T} + \frac{1}{Lp^2}\right)^b$	✗
this work	$\tilde{\mathcal{O}}\left(\sqrt{\frac{L\sigma^2}{nT}} + \left(\frac{L\sigma}{pT}\right)^{2/3} + \frac{L}{p^2 T}\right)$	✓

[a] The constants C_1 and C_2 depend on the graph parameter p, smoothness L and number of nodes n.
[b] Note that the second term is not decreasing with T.

$$\mathcal{O}\left(\frac{\sigma^2}{\mu nT} + \frac{L\left(\zeta^2 + p\sigma^2\right)}{\mu^2 p^2 T^2} + \frac{LR_0^2}{p}\exp\left[-\frac{\mu pT}{L}\right]\right). \tag{D-SGD}$$

While GT is agnostic to data-heterogenity, here the convergence estimate depends on the data-heterogenity, measured by a constant ζ^2 that satisfies:

$$\tfrac{1}{n}\sum_{i=1}^n \|\nabla f_i(\mathbf{x}^\star) - \nabla f(\mathbf{x}^\star)\|_2^2 \le \zeta^2\,. \tag{15}$$

Comparing with 1, GT completely removes dependence on data heterogeneity level ζ. Moreover, even in the homogeneous case when $\zeta = 0$ GT enjoys the same rate as D-SGD for many practical graphs when c is bounded by a constant.

Comparison to D^2. Similarly to GT, D^2 also removes the dependence on functions heterogeneity. The convergence rate of D^2 holds under assumption that $\lambda_{\min}(W) > -\frac{1}{3}$ and it is equal to [48]:

$$\mathcal{O}\left(\frac{\bar{\sigma}^2}{\mu nT} + \frac{L\bar{\sigma}^2}{\mu^2 pT^2} + \frac{LR_0^2}{p}\exp\left[-\frac{\mu pT}{L}\right]\right). \tag{D^2}$$

Under the assumption $\lambda_{\min}(W) > -\frac{1}{3}$ the parameter c is a constant, and the GT rate estimated in Theorem 1 matches (D^2).

7 Experiments

In this section we illustrate the tightness of parameters c and p in our theoretical result.

Setup. We consider simple quadratic functions defined as $f_i(\mathbf{x}) = a_i\|\mathbf{x}\|^2$ with $a_i = i$. We add artificially stochastic noise to gradients as $\nabla F_i(\mathbf{x}, \xi) = \nabla f_i(\mathbf{x}) + \xi$, where $\xi \sim \mathcal{N}(0, \frac{\sigma^2}{d}I)$ so that Assumption 4 is satisfied.

We verify the dependence on graph parameters p and c for the stochastic noise term. We fix the stepsize γ to be constant, vary p and c and measure the value of $f(\bar{\mathbf{x}}^{(t)}) - f^\star$ that GT reaches after a large number of steps. According to the theory, GT converges to the level $\mathcal{O}\left(\frac{\gamma\sigma^2}{n} + \frac{\gamma^2\sigma^2}{pc^2}\right)$ in a

linear number of steps (to reach higher accuracy, smaller stepsizes must be used). To decouple the second term we need to ensure that the first term is small enough. For that, we take the number of nodes n to be large. In all experiments we ensure that the first term is at least by order of magnitude smaller than the second by comparing the noise level with GT on a fully-connected topology.

The effect of p. First, on Figure 1 we verify $\mathcal{O}\left(\frac{\gamma^2\sigma^2}{pc^2}\right)$ when c is a constant. For a fixed $n = 300$ number of nodes with number of dimensions $d = 100$ we vary the value of a parameter p by interpolating the ring topology (with uniform weights) with the fully-connected graph. The loss value $f(\mathbf{x}^{(\infty)})$ scales linearly in $\frac{1}{p}$ as can be observed in Figure 1 and the dependency on p can thus not further be improved.

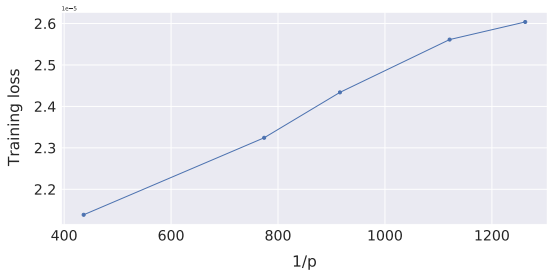

Figure 1: Impact of p on convergence with the stochastic noise $\sigma^2 = 1$, when c is kept constant. We see a linear scaling of the loss compared $\frac{1}{p}$ that verifies the term $\mathcal{O}\left(\frac{\gamma^2\sigma^2}{pc^2}\right)$.

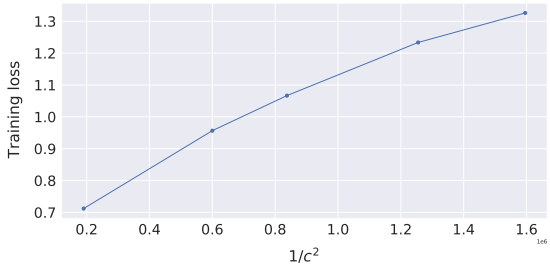

Figure 2: Impact of c on convergence with the stochastic noise $\sigma^2 = 10$. Varying c in the graph we can see quadratic dependence $\mathcal{O}\left(\frac{\gamma^2\sigma^2}{pc^2}\right)$.

The effect of c. In Figure 2 we verify the dependence of the term $\mathcal{O}\left(\frac{\gamma^2\sigma^2}{pc^2}\right)$ on parameter c. For a fixed number of $n = 300$ nodes we take the ring topology and reduce the self-weights to achieve different values of c (see appendix for details). Otherwise the setup is as above.

8 Conclusion

We have derived improved complexity bounds for the GT method, that improve over all previous results. We verify the tightness of the second term in the convergence rate in numerical experiments. Our analysis identifies that the smallest eigenvalue of the mixing matrix has a strong impact on the performance of GT, however the smallest eigenvalue can often be controlled in practice by choosing large enough self-weights (w_{ii}) on the nodes.

Our proof technique might be of independent interest in the community and might lead to improved analyses for other gossip based methods where the mixing matrix is not contracting (for e.g. in directed graphs, or using row- or column-stochastic matrices).

References

[1] Sulaiman A. Alghunaim and Ali H. Sayed. Linear convergence of primal-dual gradient methods and their performance in distributed optimization. *arXiv preprint arXiv:1904.01196*, 2019. URL https://arxiv.org/abs/1904.01196.

[2] Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Michael Rabbat. Stochastic gradient push for distributed deep learning. 2019.

[3] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Trans. Netw.*, 14(SI):2508–2530, 2006. URL https://doi.org/10.1109/TIT.2006.874516.

[4] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *J. Mach. Learn. Res.*, 13(1):165–202, January 2012. URL http://dl.acm.org/citation.cfm?id=2503308.2188391.

[5] 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, 2012. doi: 10.1109/TAC.2011.2161027.

[6] S. Gerschgorin. Uber die abgrenzung der eigenwerte einer matrix. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na*, 6:749–754, 1931. URL http://mi.mathnet.ru/izv5235.

[7] Lie He, An Bian, and Martin Jaggi. Cola: Decentralized linear learning. In *NeurIPS - Advances in Neural Information Processing Systems 31*, pages 4541–4551. 2018. URL http://papers.nips.cc/paper/7705-cola-decentralized-linear-learning.pdf.

[8] Franck Iutzeler, Pascal Bianchi, Philippe Ciblat, and Walid Hachem. Asynchronous distributed optimization using a randomized alternating direction method of multipliers. In *Proceedings of the 52nd IEEE Conference on Decision and Control, CDC*, pages 3671–3676. IEEE, 2013. URL https://doi.org/10.1109/CDC.2013.6760448.

[9] B. Johansson, M. Rabi, and M. Johansson. A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Optimization*, 20(3):1157–1170, 2010. doi: 10.1137/08073038X. URL https://doi.org/10.1137/08073038X.

[10] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[11] Sai P. Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda T. Suresh. SCAFFOLD: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378*, 2019. URL https://arxiv.org/abs/1910.06378.

[12] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based computation of aggregate information. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '03. IEEE Computer Society, 2003. URL http://dl.acm.org/citation.cfm?id=946243.946317.

[13] Anastasia Koloskova, Sebastian Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *ICML - Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 3478–3487. PMLR, 2019. URL http://proceedings.mlr.press/v97/koloskova19a.html.

[14] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U. Stich. A unified theory of decentralized sgd with changing topology and local updates, 2020.

[15] Jakub Konečnỳ, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[16] Boyue Li, Shicong Cen, Yuxin Chen, and Yuejie Chi. Communication-efficient distributed optimization in networks with gradient tracking and variance reduction. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1662–1672. PMLR, 26–28 Aug 2020. URL `http://proceedings.mlr.press/v108/li20f.html`.

[17] Tian Li, Anit Kumar Sahu, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[18] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Feddane: A federated newton-type method. *arXiv preprint arXiv:2001.01920*, 2020.

[19] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of FedAvg on non-IID data. *ICLR - International Conference on Learning Representations*, openreview, 2020.

[20] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *NIPS - Advances in Neural Information Processing Systems 30*, pages 5330–5340. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7117-can-decentralized-algorithms-outperform-centralized-algorithms-a-case-study-for-decentralized-parallel-stochastic-gradient-descent.pdf`.

[21] Tao Lin, Sai Praneeth Karimireddy, Sebastian U. Stich, and Martin Jaggi. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. *CoRR*, abs/2102.04761, 2021. URL `https://arxiv.org/abs/2102.04761`.

[22] Paolo Di Lorenzo and Gesualdo Scutari. Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, 2016. doi: 10.1109/TSIPN.2016.2524588.

[23] Songtao Lu, Xinwei Zhang, Haoran Sun, and Mingyi Hong. Gnsd: a gradient-tracking based nonconvex stochastic algorithm for decentralized optimization. In *2019 IEEE Data Science Workshop (DSW)*, pages 315–321, 2019. doi: 10.1109/DSW.2019.8755807.

[24] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS - Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[25] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2016. URL `http://arxiv.org/abs/1602.05629`.

[26] Angelia Nedić, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27, 07 2016. doi: 10.1137/16M1084316.

[27] Angelia Nedić, Alex Olshevsky, Wei Shi, and César A. Uribe. Geometrically convergent distributed optimization with uncoordinated step-sizes. In *2017 American Control Conference (ACC)*, pages 3950–3955, 2017. doi: 10.23919/ACC.2017.7963560.

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

[29] A. Nedić, S. Lee, and M. Raginsky. Decentralized online optimization with global objectives and local communication. In *2015 American Control Conference (ACC)*, pages 4497–4503, 2015.

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

[31] Angelia Nedić and Alex Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2014.

[32] Lam M. Nguyen, Phuong Ha Nguyen, Peter Richtárik, Katya Scheinberg, Martin Takáč, and Marten van Dijk. New convergence aspects of stochastic gradient algorithms. *arXiv preprint arXiv:1811.12403*, 2018. URL `https://arxiv.org/abs/1811.12403`.

[33] Shi Pu and Angelia Nedić. Distributed stochastic gradient tracking methods, 2020.

[34] Guannan Qu and Na Li. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, PP:1–1, 04 2017. doi: 10.1109/TCNS.2017.2698261.

[35] M. Rabbat. Multi-agent mirror descent for decentralized stochastic optimization. In *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 517–520, 2015.

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

[37] Sebastian U. Stich. Unified optimal analysis of the (stochastic) gradient method. *arXiv preprint arXiv:1907.04232*, 2019. URL `https://arxiv.org/abs/1907.04232`.

[38] Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. In *NeurIPS - Advances in Neural Information Processing Systems 31*, pages 7663–7673. Curran Associates, Inc., 2018. URL `http://papers.nips.cc/paper/7992-communication-compression-for-decentralized-training.pdf`.

[39] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D^2: Decentralized training over decentralized data. In *ICML - Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4848–4856. PMLR, 2018. URL `http://proceedings.mlr.press/v80/tang18a.html`.

[40] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D^2: Decentralized training over decentralized data, 2018.

[41] Hanlin Tang, Xiangru Lian, Shuang Qiu, Lei Yuan, Ce Zhang, Tong Zhang, and Ji Liu. Deepsqueeze: Decentralization meets error-compensated compression. *arXiv preprint arXiv:1907.07346*, 2019. URL `https://arxiv.org/abs/1907.07346`.

[42] John N. Tsitsiklis. *Problems in decentralized decision making and computation*. PhD thesis, Massachusetts Institute of Technology, 1984.

[43] Isidoros Tziotis, Constantine Caramanis, and Aryan Mokhtari. Second order optimality in decentralized non-convex optimization via perturbed gradient tracking. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21162–21173. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/f1ea154c843f7cf3677db7ce922a2d17-Paper.pdf`.

[44] Jianyu Wang, Anit Kumar Sahu, Zhouyi Yang, Gauri Joshi, and Soummya Kar. MATCHA: speeding up decentralized SGD via matching decomposition sampling. *arXiv preprint arXiv:1905.09435*, 2019. URL `http://arxiv.org/abs/1905.09435`.

[45] E. Wei and A. Ozdaglar. Distributed alternating direction method of multipliers. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 5445–5450, 2012.

[46] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004. URL `http://www.sciencedirect.com/science/article/pii/S0167691104000398`.

[47] Ran Xin, Anit Kumar Sahu, Usman A. Khan, and Soummya Kar. Distributed stochastic optimization with gradient tracking over strongly-connected networks. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 8353–8358, 2019. doi: 10.1109/CDC40024.2019.9029217.

[48] Kun Yuan and Sulaiman A. Alghunaim. Removing data heterogeneity influence enhances network topology dependence of decentralized sgd, 2021.

[49] Kun Yuan, Bicheng Ying, Xiaochuan Zhao, and Ali H. Sayed. Exact diffusion for distributed optimization and learning - part I: algorithm development. *IEEE Trans. Signal Process.*, 67(3):708–723, 2019. doi: 10.1109/TSP.2018.2875898. URL `https://doi.org/10.1109/TSP.2018.2875898`.

[50] Kun Yuan, Bicheng Ying, Xiaochuan Zhao, and Ali H. Sayed. Exact diffusion for distributed optimization and learning - part II: convergence analysis. *IEEE Trans. Signal Process.*, 67(3):724–739, 2019. doi: 10.1109/TSP.2018.2875883. URL `https://doi.org/10.1109/TSP.2018.2875883`.

[51] Kun Yuan, Wei Xu, and Qing Ling. Can primal methods outperform primal-dual methods in decentralized dynamic optimization?, 2020.

[52] Kun Yuan, Yiming Chen, Xinmeng Huang, Yingya Zhang, Pan Pan, Yinghui Xu, and Wotao Yin. Decentlam: Decentralized momentum SGD for large-batch deep training. *CoRR*, abs/2104.11981, 2021. URL `https://arxiv.org/abs/2104.11981`.

[53] Jiaqi Zhang and Keyou You. Decentralized stochastic gradient tracking for non-convex empirical risk minimization, 2020.

[54] Minghui Zhu and Sonia Martínez. Discrete-time dynamic average consensus. *Automatica*, 46(2):322–329, 2010. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2009.10.021. URL `https://www.sciencedirect.com/science/article/pii/S0005109809004828`.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See section 6 and 8.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Sections 3–4.
   (b) Did you include complete proofs of all theoretical results? [Yes] See appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Supplementary.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Main text and appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]