# $D^2$: Decentralized Training over Decentralized Data

**Hanlin Tang** [1] **Xiangru Lian** [1] **Ming Yan** [2 3] **Ce Zhang** [4] **Ji Liu** [5 1]

## Abstract

While training a machine learning model using multiple workers, each of which collects data from its own data source, it would be useful when the data collected from different workers are *unique* and *different*. Ironically, recent analysis of decentralized parallel stochastic gradient descent (D-PSGD) relies on the assumption that the data hosted on different workers are *not too different*. In this paper, we ask the question: *Can we design a decentralized parallel stochastic gradient descent algorithm that is less sensitive to the data variance across workers?* In this paper, we present $D^2$, a novel decentralized parallel stochastic gradient descent algorithm designed for large data variance among workers (imprecisely, "decentralized" data). The core of $D^2$ is a variance reduction extension of D-PSGD. It improves the convergence rate from $O\left(\frac{\sigma}{\sqrt{nT}} + \frac{(n\zeta^2)^{\frac{1}{3}}}{T^{2/3}}\right)$ to $O\left(\frac{\sigma}{\sqrt{nT}}\right)$ where $\zeta^2$ denotes the variance among data on different workers. As a result, $D^2$ is robust to data variance among workers. We empirically evaluated $D^2$ on image classification tasks, where each worker has access to only the data of a limited set of labels, and find that $D^2$ significantly outperforms D-PSGD.

## 1. Introduction

Training machine learning models in a decentralized way has attracted intensive interests recently (Lian et al., 2017a; Yuan et al., 2016; Colin et al., 2016). In the decentralized setting, there is a set of workers, each of which collects data from different data sources. Instead of sending all data to a centralized place, these workers only communicate with their *neighbors*. The goal is to get a model that is the same as if all data are collected in a centralized place. Decentralized learning algorithms are important in scenarios where the centralized communication is expensive or impossible, or the underlying communication network has high latency.

For decentralized learning to provide benefits, each user should provide data that is somehow *unique*, i.e., the variance of data collected from different workers are large. However, many recent theoretical results (Lian et al., 2017a;b; Nedic & Ozdaglar, 2009; Yuan et al., 2016) assume a bounded data variance across workers — when data hosted on different workers are very different, these approaches converge slowly, both empirically and theoretically. In this paper, we aim at bringing this discrepancy between the current theoretical understanding and the requirements from *some* practical scenarios.

In this paper, we present $D^2$, a novel decentralized learning algorithm designed to be robust under high data variance. $D^2$ is built upon decentralized parallel stochastic gradient descent (D-PSGD), but benefits from an additional variance reduction component. In $D^2$, each worker stores the stochastic gradient and its local model in the previous iterate and linearly combines them with the current stochastic gradient and local model. It results in an improved convergence rate over D-PSGD by eliminating the data variation among workers. In particular, the convergence rate is improved from $O\left(\frac{\sigma}{\sqrt{nT}} + \frac{(n\zeta^2)^{\frac{1}{3}}}{T^{2/3}}\right)$ to $O\left(\frac{\sigma}{\sqrt{nT}}\right)$ where $\zeta^2$ is the data variation among all workers, $\sigma^2$ is the data variance within each worker, $n$ is the number of workers, and $T$ is the number of iterations. We empirically show $D^2$ can significantly outperform D-PSGD by training an image classification model where each worker has access to only the data of a limited set of labels.

Throughout this paper, we consider the following decentralized optimization:

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} \quad f(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} \overbrace{\mathbb{E}_{\xi\sim\mathcal{D}_i} F_i(\boldsymbol{x};\xi)}^{=:f_i(\boldsymbol{x})}, \tag{1}$$

[*]Equal contribution [1]Department of Computation Science, University of Rochester [2]Department of Computational Mathematics, Science and Engineering, Michigan State University [3]Department of Mathematics, Michigan State University [4]Department of Computer Science, ETH Zurich [5]Tencent AI Lab. Correspondence to: Hanlin Tang <htang14@ur.rochester.edu>, Xiangru Lian <xiangru@yandex.com>, Ming Yan <yanm@math.msu.edu>, Ce Zhang <ce.zhang@inf.ethz.ch>, Ji Liu <ji.liu.uwisc@gmail.com>.

*Proceedings of the $35^{th}$ International Conference on Machine Learning*, Stockholm, Sweden, PMLR 80, 2018. 

where $n$ is the number of workers and $\mathcal{D}_i$ is the local data distribution for worker $i$. All workers are connected through a connected graph. Each worker can only exchange information with its neighbors.

**Definitions and notation** Throughout this paper, we use following notation and definitions:

- $\|\cdot\|_F$ denotes the Frobenius norm of matrices.
- $\|\cdot\|$ denotes the $\ell_2$ norm for vectors and the spectral norm for matrices.
- $\nabla f(\cdot)$ denotes the gradient of a function $f$.
- $f^*$ denotes the optimal solution of (1).
- $\lambda_i(\cdot)$ denotes the $i$th largest eigenvalue of a matrix.
- $\boldsymbol{x}^{(i)}$ denotes the local model of worker $i$.
- $\nabla F_i(\boldsymbol{x}^{(i)};\xi^{(i)})$ denotes a local stochastic gradient of worker $i$.
- $\mathbf{1}=[1,1,\cdots,1]^\top\in\mathbb{R}^n$ denotes the all-one vector.
- In order to organize the algorithm more clearly, here we define the concatenation of all local variables, stochastic gradients, and their averages respectively:

$$
\begin{aligned}
X :=&[\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(n)}]\in\mathbb{R}^{N\times n},\\
\overline{X} :=&X\frac{\mathbf{1}}{n}=\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}^{(i)},\\
G(X;\xi) :=&[\nabla F_1(\boldsymbol{x}^{(1)};\xi^{(1)}),\ldots,\nabla F_n(\boldsymbol{x}^{(n)};\xi^{(n)})]\\
&\in\mathbb{R}^{N\times n},\\
\overline{G}(X,\xi) :=&G(X,\xi)\frac{\mathbf{1}}{n}=\frac{1}{n}\sum_{i=1}^{n}\nabla F_i(\boldsymbol{x}^{(i)};\xi^{(i)}),\\
\nabla f(\overline{X}) :=&\sum_{i=1}^{n}\frac{1}{n}\nabla f_i\left(\overline{X}\right),\\
\overline{\nabla f}(X) :=&\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x^{(i)}),
\end{aligned}
$$

  where $\xi$ is the collection of randomly sampled data from all workers.

**Organization** This paper is organized as follows: Section 2 reviews related work about the proposed approach; Section 3 introduces the state-of-the-art decentralized stochastic gradient descent method and its convergence rate; Section 4 introduces the proposed algorithm and its intuition why it improves the state-of-the-art approach; Section 5 provides the theoretical guarantee; and Section 6 validates the proposed approaches via empirical study; and Section 7 concludes this paper.

## 2. Related work

In this section, we review the stochastic gradient descent algorithm and its decentralized variants, decentralized algorithms, and previous variance reduction technologies.

**Stochastic gradient descent (SGD)** The SGD approahces (Ghadimi & Lan, 2013; Moulines & Bach, 2011; Nemirovski et al., 2009) is quite powerful for solving large-scale machine learning problems. It achieves a convergence rate of $O\left(1/\sqrt{T}\right)$. As an implementation of SGD, the *Centralized Parallel Stochastic Gradient Descent* (**C-PSGD**), has been widely used in parallel computation. In C-PSGD, a central worker, whose job is to perform the variable updates, is connected to many leaf workers that are used to compute stochastic gradients in parallel. C-PSGD has been applied to many deep learning frameworks, such as CNTK (Seide & Agarwal, 2016), MXNet (Chen et al., 2015), and TensorFlow (Abadi et al., 2016). The convergence rate of C-PSGD is $O\left(\frac{1}{\sqrt{nT}}\right)$, which shows that it can achieve linear speedup with regards to the number of leaf workers.

**Decentralized algorithms** Centralized algorithms require a central server to communicate with all other workers (Suresh et al., 2017). In contrast, decentralized algorithms work on any connected network and only rely on the information exchange between neighbor workers (Kashyap et al., 2007; Lavaei & Murray, 2012; Nedic et al., 2009).

Decentralized algorithms are especially useful under a network with limited bandwidth or high latency. It is more favorable when data privacy is sensitive. These advantages have led to successful applications. The decentralized approach for multi-task reinforcement learning was studied in Omidshafiei et al. (2017); Mhamdi et al. (2017). In Colin et al. (2016), a dual based decentralized algorithm was proposed to solve the pairwise function optimization. Shi et al. (2014) and Mokhtari & Ribeiro (2015) analyzed the decentralized version of the ADMM optimization algorithm. An information theoretic approach was used to analyze decentralization in Dobbe et al. (2017). The decentralized version of (sub-)gradient descent was studied in Nedic & Ozdaglar (2009); Yuan et al. (2016). Its $O(1/\sqrt{T})$ convergence requires a diminishing stepsize or a constant stepsize that depends on the total number of iterations. This phenomenon happens because of the variance between the data in different workers, which we call "outer variance" to differentiate it from the variance in SGD. Recently, there are several deterministic decentralized optimization algorithms that allows a constant stepsize. For example, EXTRA (Shi et al., 2015a) is the first modification of decentralized gradient descent that converges under a constant stepsize. Later this algorithm is extended for problems with the sum of smooth and nonsmooth functions at each node (Shi et al., 2015b).

The algorithm DIGing is proposed in Nedić et al. (2017), where two exchanges are needed in each iteration. However, their stepsizes depend on both the Lipschitz constant of the differentiable function and the network structure. NIDS is the first algorithm that has a constant network independent stepsize (Li et al., 2017). This algorithm was simultaneously proposed by Yuan et al. (2017) for the smooth case only using a different approach.

**Decentralized parallel stochastic gradient descent (D-PSGD)** The D-PSGD algorithm (Nedic & Ozdaglar, 2009; Ram et al., 2010a;b) requires each worker to compute a stochastic gradient and exchange its local model with neighbors. In Duchi et al. (2012), a dual averaging based method is proposed for solving the constrained decentralized SGD optimization. In Yuan et al. (2016), the convergence rate for D-PSGD was analyzed when the gradient is assumed to be bounded. In Lan et al. (2017), a decentralized primal-dual type method was proposed with a computational complexity of $O\left(n/\epsilon^2\right)$ for general convex objectives. Lian et al. (2017a) proved that D-PSGD can admits linear speedup with respect to the number of workers with a similar convergence rate as C-PSGD.

**Variance reduction technology** There have been many methods developed for reducing the variance in SGD, including SVRG (Johnson & Zhang, 2013), SAGA (Defazio et al., 2014), SAG (Schmidt et al., 2017), MISO (Mairal, 2015), and mS2GD (Konečnỳ et al., 2016). However, most of these technologies are designed for centralized approaches. The DSA algorithm (Mokhtari & Ribeiro, 2016) applied the variance reduction similar to SAGA on strongly convex decentralized optimization problems and proved a linear convergence rate. However, the speedup property is unclear and a table of all stochastic gradients need to be stored.

## 3. Preliminary: decentralized stochastic gradient descent

The decentralized stochastic gradient descent (Lian et al., 2017a; Zhang et al., 2017; Shahrampour & Jadbabaie, 2017) allows each worker (say worker $i$) maintaining its own local variable $\boldsymbol{x}^{(i)}$. During each iteration (say, iteration $t$), each worker performs the following steps:

1. Query its neighbors' local variables.
2. Take weighted average with its local variable and its neighbors' local variables:
$$\boldsymbol{x}_{t+\frac{1}{2}}^{(i)} = \sum_{j=1}^{n} W_{ij}\boldsymbol{x}_t^{(j)},$$
where $W_{ij}$ is the $(i,j)$ element of the matrix $W$. $W_{ij}=0$ means worker $i$ and worker $j$ are not connected.
3. Perform one stochastic gradient descent step
$$\boldsymbol{x}_{t+1}^{(i)} = \boldsymbol{x}_{t+\frac{1}{2}}^{(i)} - \gamma\nabla F(\boldsymbol{x}_t^{(i)};\xi_t^{(i)}),$$
where $\xi_t^{(i)}$ represents the data sampled in worker $i$ at the iteration $t$ following the distribution $\mathcal{D}_i$.

From a global point of view, the update rule of D-PSGD can be viewed as

$$X_{t+1} = X_tW - \gamma G(X_t;\xi_t).$$

It admits the following rate shown in Theorem 1.

**Theorem 1** (Convergence rate of D-PSGD (Lian et al., 2017a))**.** *Under certain assumptions, the output of D-PSGD admits the following inequality*

$$\frac{1-\gamma L}{2T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\overline{\nabla f}(X_t)\right\|^2 + \frac{D_1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\overline{X}_t\right)\right\|^2$$
$$\leq \frac{f(0)-f^*}{\gamma T} + \frac{\gamma L}{2n}\sigma^2 + \frac{\gamma^2L^2n\sigma^2}{(1-\lambda)D_2} + \frac{9\gamma^2L^2n\varsigma^2}{(1-\sqrt{\lambda})^2D_2},$$

*where $\rho$ reflects the property of the network, $D_1$ and $D_2$ are defined to be*

$$D_1 := \left(\frac{1}{2} - \frac{9\gamma^2L^2n}{(1-\sqrt{\rho})^2D_2}\right),$$
$$D_2 := \left(1 - \frac{18\gamma^2}{(1-\sqrt{\rho})^2}nL^2\right),$$

*and $\sigma$ and $\varsigma$ measure the variation within each worker and among all workers respectively*

$$\mathbb{E}_{\xi\sim\mathcal{D}_i}\left\|\nabla F_i(\boldsymbol{x};\xi) - \nabla f_i(\boldsymbol{x})\right\|^2 \leqslant \sigma^2, \quad \forall i, \forall \boldsymbol{x}, \tag{2}$$
$$\frac{1}{n}\sum_{i=1}^{n}\left\|\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\right\|^2 \leqslant \zeta^2, \quad \forall i, \forall \boldsymbol{x}. \tag{3}$$

Choosing the optimal steplength $\gamma = \frac{1}{L+\sigma\sqrt{\frac{K}{n}}+n^{\frac{1}{3}}\zeta^{\frac{2}{3}}T^{\frac{1}{3}}}$ we have the following convergence rate:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}(\|\nabla f(\overline{X}_t)\|^2) \leq O\left(\frac{\sigma}{\sqrt{nT}} + \frac{n^{\frac{1}{3}}\zeta^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{1}{T}\right).$$

The proposed $D^2$ algorithm can improve the convergence rate by removing the dependence to the global bound of outer variance $\zeta$.

**Algorithm 1** The $D^2$ algorithm

**Input:** Initial point $\boldsymbol{x}_0^{(i)} = \mathbf{0}$, step length $\gamma > 0$, confusion matrix $W$, and the total number of iterations $T$.
**for** t = 0,1,2,. . . ,T **do**
  Randomly sample $\xi_t^{(i)}$ from the local data of the $i$th worker.
  Compute a local stochastic gradient based on $\xi_t^{(i)}$ and current variable $\boldsymbol{x}_t^{(i)}$ : $\nabla F_i(\boldsymbol{x}_t^{(i)}; \xi_t^{(i)})$.
  **if** t=0 **then**
    $\boldsymbol{x}_{t+\frac{1}{2}}^{(i)} = \boldsymbol{x}_t^{(i)} - \gamma \nabla F_i(\boldsymbol{x}_t^{(i)}; \xi_t^{(i)})$,
  **else**
    $\boldsymbol{x}_{t+\frac{1}{2}}^{(i)} = 2\boldsymbol{x}_t^{(i)} - \boldsymbol{x}_{t-1}^{(i)} - \gamma \nabla F_i(\boldsymbol{x}_t^{(i)}; \xi_t^{(i)}) + \gamma \nabla F_i(\boldsymbol{x}_{t-1}^{(i)}; \xi_{t-1}^{(i)})$.
  **end if**
  Each worker sends $\boldsymbol{x}_{t+\frac{1}{2}}^{(i)}$ to its neighbors and takes the weighted average

$$\boldsymbol{x}_{t+1}^{(i)} = \sum_{j=1}^{n} W_{ij} \boldsymbol{x}_{t+\frac{1}{2}}^{(j)},$$

  where $\boldsymbol{x}_{t+\frac{1}{2}}^{(j)}$ is from the worker $j$.
**end for**
**Output:** $\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_T^{(i)}$

## 4. The $D^2$ algorithm

In $D^2$ algorithm, each worker (say, worker $i$) repeats the following updating rule (say, at iteration $t$):

1. Compute a local stochastic gradient $\nabla F(\boldsymbol{x}_t^{(i)}; \xi_t^{(i)})$ by sampling $\xi_t^{(i)}$ from distribution $\mathcal{D}^{(i)}$;

2. Update the local model $\boldsymbol{x}_{t+\frac{1}{2}}^{(i)} \leftarrow 2\boldsymbol{x}_t^{(i)} - \boldsymbol{x}_{t-1}^{(i)} - \gamma \nabla F_i\left(\boldsymbol{x}_t^{(i)}; \xi_t^{(i)}\right) + \gamma \nabla F_i\left(\boldsymbol{x}_{t-1}^{(i)}; \xi_{t-1}^{(i)}\right)$ using the local models and stochastic gradients in both the $t$th iteration and the $(t-1)$th iteration.

3. When the synchronization barrier is met, exchange $\boldsymbol{x}_{t+\frac{1}{2}}^{(i)}$ with neighbors:

$$\boldsymbol{x}_{t+1}^{(i)} = \sum_{j=1}^{n} W_{ij} \boldsymbol{x}_{t+\frac{1}{2}}^{(j)}.$$

From a global point of view, the update rule of $D^2$ can be viewed as:

$$X_{t+1} = (2X_t - X_{t-1} - \gamma G(X_t; \xi_t) + \gamma G(X_{t-1}; \xi_{t-1}))\, W.$$

The complete algorithm is summarized in Algorithm 1.

**$D^2$ essentially runs the stochastic gradient descent step.** To understand the intuition of $D^2$, let us consider the mean value $\overline{X}_t$, which is updated just like the standard stochastic gradient descent:

$$\begin{aligned}
\overline{X}_{t+1} =& (2X_t - X_{t-1} - \gamma G(X_t; \xi_t) + \gamma G(X_{t-1}; \xi_{t-1}))\, W \frac{\mathbf{1}_n}{n}, \\
\overline{X}_{t+1} =& 2\overline{X}_t - \overline{X}_{t-1} - \gamma \overline{G}(X_t; \xi_t) + \gamma \overline{G}(X_{t-1}; \xi_{t-1}),
\end{aligned}$$

or equivalently

$$\begin{aligned}
&\overline{X}_{t+1} - \overline{X}_t \\
=&\overline{X}_t - \overline{X}_{t-1} - \gamma \overline{G}(X_t; \xi_t) + \gamma \overline{G}(X_{t-1}; \xi_{t-1}), \\
=&\overline{X}_1 - \overline{X}_0 - \gamma \sum_{k=1}^{t} \left(\overline{G}(X_t; \xi_t) - \overline{G}(X_{t-1}; \xi_{t-1})\right) \\
=& - \gamma \overline{G}(X_t; \xi_t). \quad (X_1 = X_0 - \gamma G(X_0; \xi_0)).
\end{aligned} \tag{4}$$

**Why $D^2$ improves the D-PSGD?** We may notice that D-PSGD also essentially updates in the form of stochastic gradient descent in (4). Then why D$^2$ can improve D-PSGD?

Assume that $X_t$ has achieved the optimum $X^* := x^* \mathbf{1}^\top$ with all local models equal to the optimum $x^*$ to (1). Then for D-PSGD, the next update will be

$$X_{t+1} = X^* - \gamma G(X^*; \xi_t).$$

It shows that the convergence when we approach a solution is affected by $\mathbb{E}[\|G(X^*; \xi_t\|_F^2]$, which is bounded by

$$\mathcal{O}(\sigma^2 + \zeta^2),$$

as we can see from the following:

$$\begin{aligned}
&\mathbb{E}[\|G(X^*; \xi_t\|_F^2] \\
=&\mathbb{E}\sum_{i=1}^{n} \left\| \left(\nabla F_i(\boldsymbol{x}^*; \xi_{t+1}^{(i)}) - \nabla f_i(\boldsymbol{x}^*)\right) + \nabla f_i(\boldsymbol{x}^*) \right\|^2 \\
\leq&2\mathbb{E}\sum_{i=1}^{n} \left\| \left(\nabla F_i(\boldsymbol{x}^*; \xi_{t+1}^{(i)}) - \nabla f_i(\boldsymbol{x}^*)\right) \right\|^2 \\
&+ 2\left\|\nabla f_i(\boldsymbol{x}^*) - \nabla f(\boldsymbol{x}^*)\right\|^2 \\
\leq&2\sigma^2 + 2\zeta^2.
\end{aligned}$$

Next we apply a similar analysis for $D^2$ by assuming that both $X_{t-1}$ and $X_t$ have reached the optimal solution $X^*$. The next update for $D^2$ will be:

$$X_{t+1} = (X^* - \gamma G(X^*; \xi_t) - \gamma G(X^*; \xi_{t-1}))\, W.$$

It shows that for $D^2$, the convergence when we approach a solution relies on the magnitude of $\mathbb{E}[\|G(X^*; \xi_t) - G(X^*; \xi_{t-1})\|_F^2]$, which is bounded by:

$$\mathcal{O}(\sigma^2),$$

which can be seen from:

$$\begin{aligned}
&\mathbb{E}[\|G(X^*;\xi_t)-G(X^*;\xi_{t-1})\|_F^2\\
=&\mathbb{E}\sum_{i=1}^{n}\left\|\nabla F_i(\boldsymbol{x}^*;\xi_t^{(i)})-\nabla f_i(\boldsymbol{x}^*)\right\|^2\\
&-\mathbb{E}\sum_{i=1}^{n}\left\|\nabla F_i(\boldsymbol{x}^*;\xi_{t-1}^{(i)})-\nabla f_i(\boldsymbol{x}^*)\right\|^2\\
\leq&2\sigma^2.
\end{aligned}$$

## 5. Theoretical guarantee

This section provides the theoretical guarantee for the proposed $D^2$ algorithm. We first give the assumptions required below.

**Assumption 1.** *Throughout this paper, we make the following commonly used assumptions:*

1. ***Lipschitzian gradient:*** *All function $f_i(\cdot)$'s are with L-Lipschitzian gradients.*

2. ***Bounded variance:*** *Assume bounded variance of stochastic gradient* within *each worker*

$$\mathbb{E}_{\xi\sim\mathcal{D}_i}\left\|\nabla F_i(\boldsymbol{x};\xi)-\nabla f_i(\boldsymbol{x})\right\|^2\leqslant\sigma^2,\quad\forall i,\forall\boldsymbol{x}.$$

3. ***Symmetric confusion matrix:*** *The confusion matrix $W$ is symmetric and satisfies $W\mathbf{1}=\mathbf{1}$.*

4. ***Spectral gap:*** *Let the eigenvalues of $W\in\mathbb{R}^{n\times n}$ be $\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_n$. Denote by for short*

$$\lambda:=\max_{i\in\{2,\cdots,n\}}\lambda_i=\lambda_2.$$

*We assume $\lambda<1$ and $\lambda_n>-\frac{1}{3}$.*

5. ***Initialization:*** *W.l.o.g., assume all local variables are initialized by zero, that is, $X_0=0$.*

Existing decentralized consensus algorithms (Shi et al., 2015b; Li et al., 2017) use a modification of the doubly stochastic matrix such that $\lambda>0$, i.e., choose $W=(\tilde{W}+I)/2$ where $W$ is a doubly stochastic matrix. Recently, Li & Yan (2017) show that $\lambda_n>-1/3$ is optimal in the convergence of EXTRA. However, the optimal $\lambda_n$ for NIDS (Li et al., 2017) is unknown. In this paper, we proved that $-\frac{1}{3}$ is the infimum of $\lambda_n$, and when it reduces to deterministic case, this condition is weaker than that in (Li et al., 2017). This is important, because we actually can use a $W$ that performs better.

Given Assumption 1, we have following convergence guarantee for $D^2$:

**Theorem 2** (Convergence of Algorithm 1)**.** *Choose the steplength $\gamma$ in Algorithm 1 to be a constant satisfying $1-24C_2\gamma^2L^2>0$. Under Assumption 1, we have the following convergence rate for Algorithm 1:*

$$\begin{aligned}
&A_1\|\nabla f(\mathbf{0})\|^2+\sum_{t=1}^{T-1}\left(\mathbb{E}\|\nabla f(\overline{X}_t)\|^2+A_2\mathbb{E}\|\overline{\nabla f}(X_t)\|^2\right)\\
\leq&\frac{2(f(0)-f^*)}{\gamma}+\frac{LT\gamma}{n}\sigma^2+\frac{6L^2C_1\gamma^2\zeta_0^2}{C_3}\\
&+\frac{12L^2C_2\gamma^2\sigma^2T}{C_3}+\frac{6L^2C_2\gamma^4L^2\sigma^2T}{nC_3}+\frac{6L^2C_1\gamma^2\sigma^2}{C_3},
\end{aligned}\tag{5}$$

*where*

$$\begin{aligned}
\zeta_0:=&\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(\mathbf{0})-\nabla f(\mathbf{0})\|^2,\\
v:=&\lambda_n-\sqrt{\lambda_n^2-\lambda_n},\\
C_1:=&\max\left\{\frac{1}{1-|v|^2},\frac{1}{(1-\lambda)^2}\right\}\geq1,\\
C_2:=&\max\left\{\frac{\lambda_n^2}{(1-|v|^2)},\frac{\lambda^2}{(1-\sqrt{\lambda})^2(1-\lambda)}\right\},\\
C_3:=&1-24C_2\gamma^2L^2,\\
A_1:=&1-\frac{6L^2C_1\gamma^2}{C_3},\\
A_2:=&1-L\gamma-\frac{6L^2C_2\gamma^4L^2}{C_3}.
\end{aligned}$$

By appropriately specifying the step length $\gamma$, we reach the following corollary:

**Corollary 3.** *Choose the step length $\gamma$ in Algorithm 1 to be $\gamma=\frac{1}{8\sqrt{C_2}L+6\sqrt{C_1}L+\sigma\sqrt{\frac{T}{n}}}$, where $C_1$ and $C_2$ are defined in Theorem 2. Under Assumption 1, the following convergence rate holds*

$$\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\|\nabla f(\overline{X_t})\|^2\lesssim&\frac{\sigma}{\sqrt{nT}}+\frac{1}{T}+\frac{\zeta_0^2}{T+\sigma^2T^2}\\
&+\frac{\sigma^2}{1+\sigma^2T},
\end{aligned}$$

*where $\zeta_0$ is defined in Theorem 2 and we treat $f(0)-f^*$, $L$, $\lambda_n$, and $\lambda$ as constants.*

Note that we can obtain even better constants by choosing different parameters and applying tighter inequalities, however, the main result of this corollary is to show the order of the convergence. We highlight a few key observations from our theoretical results in the following.

**Tightness of the convergence rate** Setting $\sigma=0$ and $\zeta_0=0$, which reduces the VR-SGD to a normal GD algorithm, we shall see that the convergence rate becomes $O\left(\frac{1}{T}\right)$, which is exactly the rate of GD.

**Linear speedup** Since the leading term of the convergence rate is $O\left(\frac{1}{\sqrt{nT}}\right)$, which is consistent with the convergence rate of C-PSGD, this indicates that we would achieve a linear speed up with respect to the number of nodes.

**Consistent with NIDS** In NIDS (Li & Yan, 2017), the term depends on $\zeta_0$ in the convergence rate is $O\left(\frac{\zeta_0^2}{T}\right)$. While the corresponding term in $D^2$ is $O\left(\frac{\zeta_0^2}{T+\sigma^2T^2}\right)$, which indicates when our algorithm is consistent with NIDS because in NIDS $\sigma$ is considered to be 0.

**Superiority over D-PSGD** When compared to D-PSGD, the convergence rate of $D^2$ only depends on $\zeta_0$, and the corresponding decaying rate is $\frac{\zeta_0}{T^2}$. Whereas in D-PSGD (Lian et al., 2017a), we need to assume an upper bound for the global variance between different nodes' dataset, and its influence can be compared to $\sigma^2$, the inner variance of each node itself. This means we can always achieve a much better convergence rate than D-PSGD.

## 6. Experiments

We evaluate the effectiveness of D$^2$ by comparing it with both centralized and decentralized SGD algorithms.

### 6.1. Experiment Settings

We conduct experiments in two settings.

1. TRANSFERLEARNING: We test the case that each worker has access to a local pre-trained neural network as feature extractor, and we want to train a logistic regression model among all these workers. In our experiment, we select the first 16 classes of ImageNet and use InceptionV4 as the feature extractor to extract 2048 features for each image. We conduct data augmentation and generate a blurblack version for each image. In total this dataset contains $16\times1300\times2$ images.

2. LENET: We test the case that all workers collaboratively train a neural network model. We train a LeNet on the CIFAR10 dataset. In total this dataset contains 50,000 images of size $32\times32$.

One caveat of training more recent neural networks is that modern architectures often have a batch normalization layer, which inherently assumes that the data distribution is uniform across different batches, which is not the case that we are interested in. In principle, we could also flow the batch information through the network in a decentralized way; however, we leave this as future work.

By default, each worker only has *exclusive* access to a subset of classes. For TRANSFERLEARNING, we use 16 workers and each worker has access to one class; for LENET, we use 5 workers and each worker has access to two classes. For comparison, we also consider a case when the datasets is first shuffled and then uniformly partitioned among all the workers, we call this the *shuffled case*, and the default one the *unshuffled case*. We use a ring topology for both experiments.

**Parameter Tuning.** For TRANSFERLEARNING, we use constant learning rates and tune it from {0.01, 0.025, 0.05, 0.075, 0.1}. For LENET, we use constant learning rate 0.05 which is tuned from {0.5, 0.1, 0.05, 0.01} for centralized algorithms and batch size 128 on each worker.

**Metrics.** In this paper, we mainly focus on the convergence rate of different algorithms instead of the wall clock speed. This is because the implementation of D$^2$ is a minor change over the standard D-PSGD algorithm, and thus they has almost the same speed to finish one epoch of training, and both are no slower than the centralized algorithm. When the network has high latency, if a decentralized algorithm ($D^2$ or D-PSGD) converges with a similar speed as the centralized algorithm, it can be up to one order of magnitude faster (Lian et al., 2017a). However, the convergence rate depending on the "outer variance" is different for both algorithms.

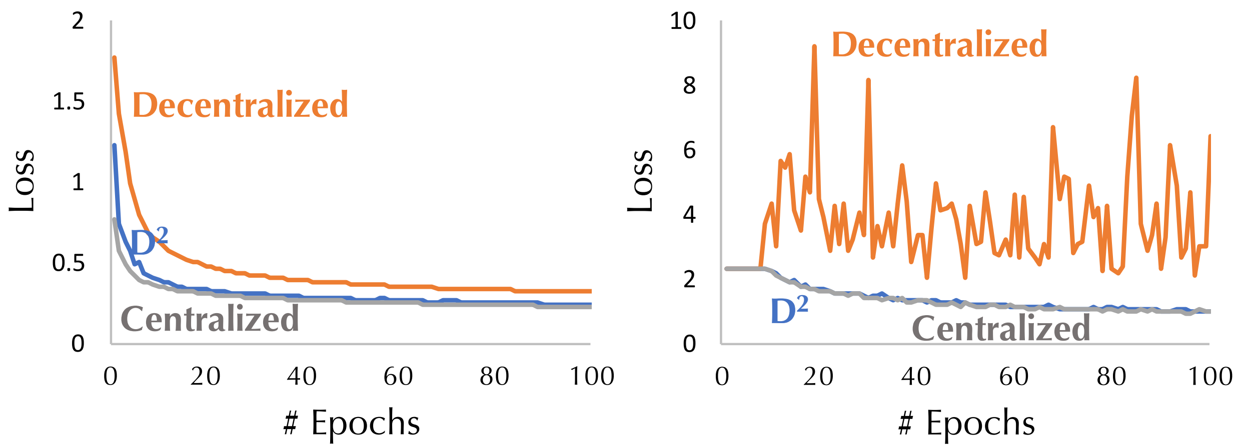
### 6.2. Unshuffled Case

variation across workers is maximized. Figure 1 shows the result. In the unshuffled case, we see that D-PSGD converges slower than the centralized case. This is consistent with the original D-PSGD paper (Lian et al., 2017a). On the other hand, D$^2$ converges much faster than D-PSGD, and achieves almost the same loss as the centralized algorithm. For the LeNet case, each worker only has access to data of assigned two labels, which means the data variation is very large. The D-PSGD does not converge with the given learning rate 0.05.[1]

### 6.3. Shuffled Case

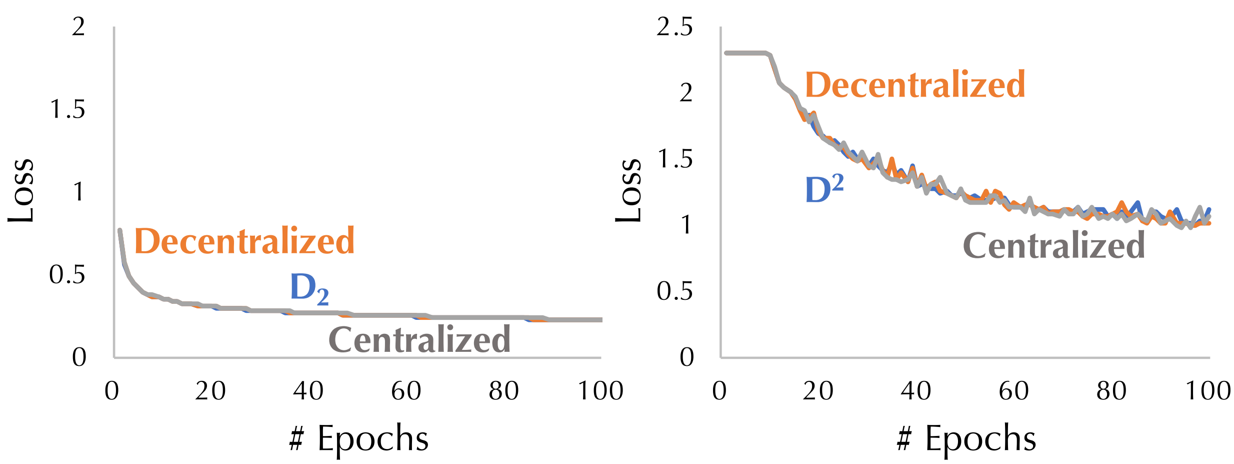
As a sanity check, Figure 2 shows the result of three different algorithms on the shuffled data. In this case, the data variation among workers is small (in expectation, they are drawn from the same distribution). We see that, all strategies have similar convergence rate. This validate that D$^2$ is more effective for larger data variation between different workers.

[1]We can tune the learning rate 50x smaller for D-PSGD to converge in this case, but doing so will make D-PSGD stuck at the starting point for quite a long time.




**(a) TRANSFERLEARNING** **(b) LENET**

*Figure 1.* Convergence of Different Distributed Training Algorithms (Unshuffled Case).




**(a) TRANSFERLEARNING** **(b) LENET**

*Figure 2.* Convergence of Different Distributed Training Algorithms (Shuffled Case).

## 7. Conclusion

In this paper, we propose a decentralized algorithm, namely, $D^2$ algorithm. $D^2$ algorithm integrates the D-PSGD algorithm with the variance reduction technology, by which we improves the convergence rate of D-PSGD. The variance reduction technology used in this paper is different from the commonly used ones such as SVRG and SAGA, that are designed for centralized approaches. Experiments validate the advantage of $D^2$ over D-PSGD — $D^2$ converges with a rate that is similar to centralized SGD while D-PSGD does not converge to a solution with a similar quality when the data variance is large. While being robust to large data variance among workers, the same performance benefit of D-PSGD over the centralized strategy still holds for $D^2$.

## Acknowledgements

This project is supported in part by NSF CCF1718513, NEC fellowship, IBM faculty award, NSF DMS-1621798, Swiss NSF NRP 75 407540_167266, IBM Zurich, Mercedes-Benz Research & Development North America, Oracle

Labs, Swisscom, Zurich Insurance, and Chinese Scholarship Council.

## References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pp. 265–283, 2016.

Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C., and Zhang, Z. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*, 2015.

Colin, I., Bellet, A., Salmon, J., and Clémençon, S. Gossip dual averaging for decentralized optimization of pairwise functions. In *International Conference on Machine Learning*, pp. 1388–1396, 2016.

Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pp. 1646–1654, 2014.

Dobbe, R., Fridovich-Keil, D., and Tomlin, C. Fully decentralized policies for multi-agent systems: An information theoretic approach. In *Advances in Neural Information Processing Systems*, pp. 2945–2954, 2017.

Duchi, J. C., Agarwal, A., and Wainwright, M. J. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3):592–606, 2012.

Ghadimi, S. and Lan, G. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013. doi: 10.1137/120880811.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pp. 315–323, 2013.

Kashyap, A., Başar, T., and Srikant, R. Quantized consensus. *Automatica*, 43(7):1192–1203, 2007.

Konečnỳ, J., Liu, J., Richtárik, P., and Takáč, M. Mini-batch semi-stochastic gradient descent in the proximal setting. *IEEE Journal of Selected Topics in Signal Processing*, 10 (2):242–255, 2016.

Lan, G., Lee, S., and Zhou, Y. Communication-efficient algorithms for decentralized and stochastic optimization. 01 2017.

Lavaei, J. and Murray, R. M. Quantized consensus by means of gossip algorithm. *IEEE Transactions on Automatic Control*, 57(1):19–32, 2012.

Li, Z. and Yan, M. A primal-dual algorithm with optimal stepsizes and its application in decentralized consensus optimization. *arXiv preprint arXiv:1711.06785*, 2017.

Li, Z., Shi, W., and Yan, M. A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *arXiv preprint arXiv:1704.07807*, 2017.

Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu, J. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. 05 2017a.

Lian, X., Zhang, W., Zhang, C., and Liu, J. Asynchronous decentralized parallel stochastic gradient descent. *arXiv preprint arXiv:1710.06952*, 2017b.

Mairal, J. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

Mhamdi, E., Mahdi, E., Hendrikx, H., Guerraoui, R., and Maurer, A. D. O. Dynamic safe interruptibility for decentralized multi-agent reinforcement learning. Technical report, EPFL, 2017.

Mokhtari, A. and Ribeiro, A. Decentralized double stochastic averaging gradient. In *Signals, Systems and Computers, 2015 49th Asilomar Conference on*, pp. 406–410. IEEE, 2015.

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

Moulines, E. and Bach, F. R. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 24*, pp. 451–459. Curran Associates, Inc., 2011.

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

Nedic, A., Olshevsky, A., Ozdaglar, A., and Tsitsiklis, J. N. On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11): 2506–2517, 2009.

Nedić, A., Olshevsky, A., and Rabbat, M. G. Network topology and communication-computation tradeoffs in decentralized optimization. *arXiv preprint arXiv:1709.08765*, 2017.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009. doi: 10.1137/070704277.

Omidshafiei, S., Pazis, J., Amato, C., How, J. P., and Vian, J. Deep decentralized multi-task multi-agent rl under partial observability. *arXiv preprint arXiv:1703.06182*, 2017.

Ram, S. S., Nedić, A., and Veeravalli, V. V. Asynchronous gossip algorithm for stochastic optimization: Constant stepsize analysis. In *Recent Advances in Optimization and its Applications in Engineering*, pp. 51–60. Springer, 2010a.

Ram, S. S., Nedić, A., and Veeravalli, V. V. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of optimization theory and applications*, 147(3):516–545, 2010b.

Schmidt, M., Le Roux, N., and Bach, F. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

Seide, F. and Agarwal, A. Cntk: Microsoft's open-source deep-learning toolkit. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 2135–2135, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2945397.

Shahrampour, S. and Jadbabaie, A. Distributed online optimization in dynamic environments using mirror descent. *IEEE Transactions on Automatic Control*, 2017.

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

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, 2015a.

Shi, W., Ling, Q., Wu, G., and Yin, W. A proximal gradient algorithm for decentralized composite optimization. *IEEE Transactions on Signal Processing*, 63(22):6013–6023, 2015b.

Suresh, A. T., Yu, F. X., Kumar, S., and McMahan, H. B. Distributed mean estimation with limited communication. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3329–3337, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Yuan, K., Ling, Q., and Yin, W. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016. doi: 10.1137/130943170.

Yuan, K., Ying, B., Zhao, X., and Sayed, A. H. Exact diffusion for distributed optimization and learning—part i: Algorithm development. *arXiv preprint arXiv:1702.05122*, 2017.

Zhang, W., Zhao, P., Zhu, W., Hoi, S. C., and Zhang, T. Projection-free distributed online learning in networks. In *International Conference on Machine Learning*, pp. 4054–4062, 2017.

# Supplemental Materials

This supplement material includes the proofs for Theorem 2.

Because the confusion matrix $W$ is symmetric, it can be decomposed as $W = P\Lambda P^\top$, where $P = (\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n)$ is an orthogonal matrix, i.e., $P^\top P = PP^\top = I$, and $\Lambda = \text{diag}\{\lambda_1, \ldots, \lambda_n\}$ is a diagonal matrix with diagonal entries being the eigenvalues of $W$ in the nonincreasing order. Then applying the decomposition to the iteration (from $W_t$ and $W_{t-1}$ to $W_{t+1}$)

$$X_{t+1} = 2X_t W - X_{t-1}W - \gamma G(X_t;\xi_t)W + \gamma G(X_{t-1};\xi_{t-1})W$$

gives

$$X_{t+1} = 2X_t P\Lambda P^\top - X_{t-1}P\Lambda P^\top - \gamma G(X_t;\xi_t)P\Lambda P^\top + \gamma G(X_{t-1};\xi_{t-1})P\Lambda P^\top.$$

Denote $Y_t = X_t P$, $H(X_t;\xi_t) = G(X_t;\xi_t)P$, and use $y_t^{(i)}$ and $h_t^{(i)}$ to indicate the $i$-th column of $Y_t$ and $H(X_t;\xi_t)$, respectively. Then

$$Y_{t+1} = 2Y_t\Lambda - Y_{t-1}\Lambda - \gamma H(X_t;\xi_t)\Lambda + \gamma H(X_{t-1};\xi_{t-1})\Lambda, \tag{6}$$

or in the columns of $Y_t$ and $H(X_t;\xi_t)$,

$$y_{t+1}^{(i)} = \lambda_i(2y_t^{(i)} - y_{t-1}^{(i)} - \gamma h_t^{(i)} + \gamma h_{t-1}^{(i)}). \tag{7}$$

From the properties of $W$ in Assumption 1 and the decomposition, we have $\lambda_1 = 1$ and $\boldsymbol{v}_1 = \frac{1}{\sqrt{n}}(1, 1, \cdots, 1)^\top$. Therefore $y_t^{(1)} = \overline{X}_t\sqrt{n}$. For all other eigenvalues $-\frac{1}{3} < \lambda_i < 1$, the equation (7) shows that all $y_t^{(i)}$ would "decay to zero", which explains how the confusion matrix works.

**Lemma 4.** *Given two non-negative sequences $\{a_t\}_{t=1}^{\infty}$ and $\{b_t\}_{t=1}^{\infty}$ that satisfying*

$$a_t = \sum_{s=1}^{t} \rho^{t-s} b_s, \tag{8}$$

*with $\rho \in [0, 1)$, we have*

$$S_k := \sum_{t=1}^{k} a_t \le \sum_{s=1}^{k} \frac{b_s}{1-\rho},$$
$$D_k := \sum_{t=1}^{k} a_t^2 \le \frac{1}{(1-\rho)^2} \sum_{s=1}^{k} b_s^2.$$

*Proof.*

$$S_k := \sum_{t=1}^{k} a_t = \sum_{t=1}^{k}\sum_{s=1}^{t} \rho^{t-s} b_s = \sum_{s=1}^{k}\sum_{t=s}^{k} \rho^{t-s} b_s = \sum_{s=1}^{k}\sum_{t=0}^{k-s} \rho^{t} b_s \le \sum_{s=1}^{k} \frac{b_s}{1-\rho}. \tag{9}$$

$$\begin{aligned}
D_k := \sum_{t=1}^{k} a_t^2 &= \sum_{t=1}^{k}\sum_{s=1}^{t} \rho^{t-s} b_s \sum_{r=1}^{t} \rho^{t-r} b_r = \sum_{t=1}^{k}\sum_{s=1}^{t}\sum_{r=1}^{t} \rho^{2t-s-r} b_s b_r \\
&\le \sum_{t=1}^{k}\sum_{s=1}^{t}\sum_{r=1}^{t} \rho^{2t-s-r} \frac{b_s^2 + b_r^2}{2} = \sum_{t=1}^{k}\sum_{s=1}^{t}\sum_{r=1}^{t} \rho^{2t-s-r} b_s^2 \\
&\le \frac{1}{1-\rho} \sum_{t=1}^{k}\sum_{s=1}^{t} \rho^{t-s} b_s^2 \le \frac{1}{(1-\rho)^2} \sum_{s=1}^{k} b_s^2
\end{aligned} \tag{10}$$

where the last inequality holds because of (9). □

**Lemma 5.** *For any matrix $X_t \in \mathbb{R}^{N\times n}$, we have*

$$\sum_{i=2}^{n}\left\|X_t\boldsymbol{v}^{(i)}\right\|^2 \leq \sum_{i=1}^{n}\left\|X_t\boldsymbol{v}^{(i)}\right\|^2 = \|X_t\|_F^2$$
$$\sum_{i=1}^{n}\left\|X_tP^\top\boldsymbol{e}^{(i)}\right\|^2 = \left\|X_tP^\top\right\|_F^2 = \|X_t\|_F^2$$

*where $\boldsymbol{e}^{(i)} \in \mathbb{R}^{n\times 1}$ with the $i$-th component being 1 and all others being 0.*

*Proof.* From the definition of the Frobenius norm for a matrix, we have

$$\sum_{i=1}^{n}\left\|X_t\boldsymbol{v}^{(i)}\right\|^2 = \|X_tP\|_F^2 = Tr\left(X_tPP^\top X_t^\top\right) = Tr\left(X_tX_t^\top\right) = \|X_t\|_F^2.$$

Since $\left\|X_t\boldsymbol{v}^{(1)}\right\|^2 \geq 0$, so

$$\sum_{i=2}^{n}\left\|X_t\boldsymbol{v}^{(i)}\right\|^2 \leq \sum_{i=1}^{n}\left\|X_t\boldsymbol{v}^{(i)}\right\|^2 = \sum_{i=1}^{n}\|X_t\|_F^2.$$

In the same way, we have

$$\sum_{i=1}^{n}\left\|X_tP^\top\boldsymbol{e}^{(i)}\right\|^2 = \left\|X_tP^\top\right\|_F^2 = \|X_t\|_F^2.$$

The result is proved. $\square$

**Lemma 6.** *Given $\rho \in (-\frac{1}{3}, 0)\bigcup(0,1)$, for any two sequence $\{a_t\}_{t=0}^{\infty}$ and $\{b_t\}_{t=0}^{\infty}$ that satisfy*

$$\begin{aligned}
a_0 =& b_0 = 0,\\
a_1 =& b_1,\\
a_{t+1} =& \rho(2a_t - a_{t-1}) + b_t - b_{t-1}, \quad \forall t \geq 1,
\end{aligned}$$

*we have*

$$a_{t+1} = a_1\left(\frac{u^{t+1}-v^{t+1}}{u-v}\right) + \sum_{s=1}^{t}\beta_s\frac{u^{t-s+1}-v^{t-s+1}}{u-v}, \quad \forall t \geq 0,$$

*where*

$$\beta_s = b_s - b_{s-1}, \quad u = \rho + \sqrt{\rho^2-\rho}, \quad v = \rho - \sqrt{\rho^2-\rho}.$$

*More specifically, if $0 < \rho < 1$, we have*

$$a_{t+1}\sin\theta = a_1\rho^{t/2}\sin\left[(t+1)\theta\right] + \sum_{s=1}^{t}\beta_s\rho^{(t-s)/2}\sin\left[(t+1-s)\theta\right], \quad \forall t \geq 0$$

*where*

$$\beta_s = b_s - b_{s-1}, \quad \theta = \arccos(\sqrt{\rho}).$$

*Proof.* When $t = 0$, the results is easy to verify. Next we consider the case $t \geq 1$. Since

$$a_{t+1} = 2\rho a_t - \rho a_{t-1} + \beta_t,$$

We can find

$$u = \rho + \sqrt{\rho^2 - \rho}, \quad v = \rho - \sqrt{\rho^2 - \rho},$$

such that

$$a_{t+1} - ua_t = (a_t - ua_{t-1})v + \beta_t. \tag{11}$$

Note that $u$ and $v$ are complex numbers when $0 < \rho < 1$. That is

$$u = \sqrt{\rho}e^{i\theta}, \quad v = \sqrt{\rho}e^{-i\theta},$$

with $\theta = \arccos(\sqrt{\rho})$.

Recursively applying (11) gives

$$\begin{aligned}
a_{t+1} - ua_t =& (a_t - ua_{t-1})v + \beta_t = (a_{t-1} - ua_{t-2})v^2 + \beta_{t-1}v + \beta_t \\
=& (a_1 - ua_0)v^t + \sum_{s=1}^{t} \beta_s v^{t-s} \\
=& a_1 v^t + \sum_{s=1}^{t} \beta_s v^{t-s}. \quad \text{(due to } a_0 = 0)
\end{aligned}$$

Diving both sides by $u^{t+1}$, we obtain

$$\begin{aligned}
\frac{a_{t+1}}{u^{t+1}} =& \frac{a_t}{u^t} + u^{-(t+1)} \left( a_1 v^t + \sum_{s=1}^{t} \beta_s v^{t-s} \right) \\
=& \frac{a_{t-1}}{u^{t-1}} + u^{-t} \left( a_1 v^{t-1} + \sum_{s=1}^{t-1} \beta_s v^{t-1-s} \right) + u^{-(t+1)} \left( a_1 v^t + \sum_{s=1}^{t} \beta_s v^{t-s} \right) \\
=& \frac{a_1}{u} + \sum_{k=1}^{t} u^{-k-1} \left( a_1 v^k + \sum_{s=1}^{k} \beta_s v^{k-s} \right)
\end{aligned}$$

Then we multiply both sides by $u^{t+1}$ and have

$$\begin{aligned}
a_{t+1} =& a_1 u^t + \sum_{k=1}^{t} u^{t-k} \left( a_1 v^k + \sum_{s=1}^{k} \beta_s v^{k-s} \right) \\
=& a_1 u^t \left( 1 + \sum_{k=1}^{t} \left( \frac{v}{u} \right)^k \right) + u^t \sum_{k=1}^{t} \sum_{s=1}^{k} \beta_s v^{-s} \left( \frac{v}{u} \right)^k \\
=& a_1 u^t \sum_{k=0}^{t} \left( \frac{v}{u} \right)^k + u^t \sum_{s=1}^{t} \sum_{k=s}^{t} \beta_s v^{-s} \left( \frac{v}{u} \right)^k \quad \left(\text{due to } \sum_{k=1}^{t} \sum_{s=1}^{k} a_s b_k = \sum_{s=1}^{t} \sum_{k=s}^{t} a_s b_k\right) \\
=& a_1 u^t \left( \frac{1 - \left(\frac{v}{u}\right)^{t+1}}{1 - \frac{v}{u}} \right) + u^t \sum_{s=1}^{t} \beta_s v^{-s} \left( \frac{v}{u} \right)^s \frac{1 - \left(\frac{v}{u}\right)^{t-s+1}}{1 - \frac{v}{u}} \\
=& a_1 \left( \frac{u^{t+1} - v^{t+1}}{u - v} \right) + \sum_{s=1}^{t} \beta_s \frac{u^{t-s+1} - v^{t-s+1}}{u - v}.
\end{aligned}$$

When $\rho \in (0, 1)$, since $u = \sqrt{\rho}e^{i\theta}$ and $v = \sqrt{\rho}e^{-i\theta}$, we have

$$a_{t+1} = a_1 \rho^{t/2} \frac{\sin[(t+1)\theta]}{\sin\theta} + \sum_{s=1}^{t} \beta_s \rho^{(t-s)/2} \frac{\sin[(t-s+1)\theta]}{\sin\theta}.$$

The result is proved. $\square$

**Lemma 7.** *Under Assumption 1, we have*

$$
\begin{aligned}
&\left(1-24C_2\gamma^2L^2\right)\sum_{i=1}^{n}\sum_{t=0}^{T}\left\|\overline{X}_t-\boldsymbol{x}_t^{(i)}\right\|^2\\
\leq\,& 2C_1\|X_1\|_F^2+12C_2\gamma^2n\sigma^2T+6C_2\gamma^4L^2\sigma^2T+6C_2\gamma^4L^2n\sum_{t=1}^{T-1}\left\|\overline{\nabla f}(X_t)\right\|^2,
\end{aligned}
$$

*where $\gamma$, $L$, $\sigma$, $\theta$, $C_1$ and $C_2$ are defined in Theorem 2.*

*Proof.* To estimate the difference of the local models and the global mean model, we have

$$
\begin{aligned}
\sum_{i=1}^{n}\left\|\overline{X}_t-\boldsymbol{x}_t^{(i)}\right\|^2 =&\sum_{i=1}^{n}\left\|X_t\boldsymbol{e}^{(i)}-X_t\frac{\mathbf{1}_n}{n}\right\|^2=\|X_t-X_t\frac{\mathbf{1}_n\mathbf{1}_n^\top}{n}\|_F^2=\|X_tPP^\top-X_t\boldsymbol{v}_1\boldsymbol{v}_1^\top\|_F^2\\
=&\left\|X_tP\begin{pmatrix}0, & 0, & 0, & \cdots, & 0\\ 0, & 1, & 0, & \cdots, & 0\\ 0, & 0, & 1, & \cdots, & 0\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ 0, & 0, & 0, & \cdots, & 1\end{pmatrix}\right\|_F^2\\
=&\sum_{i=2}^{n}\left\|y_t^{(i)}\right\|^2,
\end{aligned}
\tag{12}
$$

where $y_t^{(i)}$ is the $i$-th column of $X_tP$. Note that we have, from (7),

$$
y_{t+1}^{(i)}=\lambda_i(2y_t^{(i)}-y_{t-1}^{(i)}-\gamma h_t^{(i)}+\gamma h_{t-1}^{(i)})=\lambda_i(2y_t^{(i)}-y_{t-1}^{(i)})+\lambda_i\beta_t^{(i)},
$$

where $\beta_t^{(i)}=-\gamma h_t^{(i)}+\gamma h_{t-1}^{(i)}$. For all $y^{(i)}$ that corresponding to $-\frac{1}{3}<\lambda_i<0$, Lemma 6 shows

$$
y_{t+1}^{(i)}=y_1^{(i)}\left(\frac{u_i^{t+1}-v_i^{t+1}}{u_i-v_i}\right)+\lambda_i\sum_{s=1}^{t}\beta_s^{(i)}\frac{u_i^{t-s+1}-v_i^{t-s+1}}{u_i-v_i},
$$

where $u_i=\lambda_i+\sqrt{\lambda_i^2-\lambda_i}$ and $v_i=\lambda_i-\sqrt{\lambda_i^2-\lambda_i}$. Therefore, we have

$$
\left\|y_{t+1}^{(i)}\right\|^2\leq 2\left\|y_1^{(i)}\right\|^2\left(\frac{u_i^{t+1}-v_i^{t+1}}{u_i-v_i}\right)^2+2\lambda_i^2\left(\sum_{s=1}^{t}\left\|\beta_s^{(i)}\right\|\left|\frac{u_i^{t-s+1}-v_i^{t-s+1}}{u_i-v_i}\right|\right)^2.
\tag{13}
$$

For $\left|\frac{u_i^{n+1}-v_i^{n+1}}{u_i-v_i}\right|$, we have

$$
\left|\frac{u_i^{n+1}-v_i^{n+1}}{u_i-v_i}\right|\leq|v_i|^n\left|\frac{u_i\left(\frac{u_i}{v_i}\right)^n-v_i}{u_i-v_i}\right|\leq|v_i|^n\quad\text{(due to } |u_i|<|v_i|\text{)}.
$$

Using (13), we obtain

$$
\left\|y_{t+1}^{(i)}\right\|^2\leq 2\left\|y_1^{(i)}\right\||v_i|^{2t}+2\lambda_i^2\left(\sum_{s=1}^{t}\left\|\beta_s^{(i)}\right\||v_i|^{t-s}\right)^2.
$$

Summing from $t=0$ to $t=T-1$ gives

$$
\sum_{t=0}^{T-1}\left\|y_{t+1}^{(i)}\right\|^2=\sum_{t=1}^{T}\left\|y_t^{(i)}\right\|^2\leq 2\left\|y_1^{(i)}\right\|\sum_{t=0}^{T-1}|v_i|^{2t}+2\lambda_i^2\sum_{t=1}^{T-1}\left(\sum_{s=1}^{t}\left\|\beta_s^{(i)}\right\||v_i|^{t-s}\right)^2.
$$

Denote $a_t = \sum_{s=1}^{t} \left\|\beta_s^{(i)}\right\| |v_i|^{t-s}$, which has the same structure as the sequence in Lemma 4. Therefore, when $\lambda_i < 0$, we have

$$\begin{aligned}\sum_{t=1}^{T} \left\|y_t^{(i)}\right\|^2 \leq & \frac{2\left\|y_1^{(i)}\right\|}{1-|v_i|^2} + \frac{2\lambda_i^2}{(1-|v_i|)^2}\sum_{t=1}^{T-1}\left\|\beta_t^{(i)}\right\|^2 \\ \leq & \frac{2\left\|y_1^{(i)}\right\|}{1-|v|^2} + \frac{2\lambda_n^2}{(1-|v|)^2}\sum_{t=1}^{T-1}\left\|\beta_t^{(i)}\right\|^2,\end{aligned} \tag{14}$$

where $v = \lambda_n - \sqrt{\lambda_n^2 - \lambda_n}$.

For all $y^{(i)}$ that satisfies $0 \leq \lambda_i < 1$, from (7) and Lemma 6, we have

$$y_{t+1}^{(i)} \sin\theta_i = y_1^{(i)} \lambda_i^{t/2} \sin\left[(t+1)\theta_i\right] + \lambda_i \sum_{s=1}^{t} \beta_s^{(i)} \lambda_i^{(t-s)/2} \sin\left[(t+1-s)\theta_i\right],$$

where $\beta_s^{(i)} = -\gamma h_s^{(i)} + \gamma h_{s-1}^{(i)}$ and $\theta_i = \arccos(\sqrt{\lambda_i})$.

Then

$$\begin{aligned}\left\|y_{t+1}^{(i)}\right\|^2 \sin^2\theta_i \leq & 2\left\|y_1^{(i)}\right\|^2 \lambda_i^t \sin^2[(t+1)\theta_i] + 2\lambda_i^2\left(\sum_{s=1}^{t}\left\|\beta_s^{(i)}\sin\left[(t+1-s)\theta_i\right]\right\|\lambda_i^{(t-s)/2}\right)^2 \\ \leq & 2\left\|y_1^{(i)}\right\|^2 \lambda_i^t + 2\lambda_i^2\left(\sum_{s=1}^{t}\|\beta_s^{(i)}\|\lambda_i^{(t-s)/2}\right)^2,\end{aligned}$$

Summing from $t=0$ to $T-1$ gives

$$\sum_{t=0}^{T-1}\left\|y_{t+1}^{(i)}\right\|^2 \sin^2\theta_i = \sum_{t=1}^{T}\left\|y_t^{(i)}\right\|^2 \sin^2\theta_i \leq 2\left\|y_1^{(i)}\right\|^2 \sum_{t=0}^{T-1}\lambda_i^t + 2\lambda_i^2 \sum_{t=1}^{T-1}\left(\sum_{s=1}^{t}\|\beta_s^{(i)}\|\lambda_i^{(t-s)/2}\right)^2$$

From Lemma 4, $\sum_{s=1}^{t}\|\beta_s^{(i)}\|\lambda_i^{(t-s)/2}$ has the same structure as the sequence in Lemma 4, so we have

$$\sum_{t=1}^{T}\left\|y_t^{(i)}\right\|^2 \sin^2\theta_i \leq \frac{2\left\|y_1^{(i)}\right\|^2}{1-\lambda_i} + \frac{2\lambda_i^2}{(1-\sqrt{\lambda_i})^2}\sum_{t=1}^{T-1}\left\|\beta_t^{(i)}\right\|^2.$$

Then $\sin^2\theta_i = 1-\lambda_i$ gives

$$\begin{aligned}\sum_{t=1}^{T}\left\|y_t^{(i)}\right\|^2 \leq & \frac{2\left\|y_1^{(i)}\right\|^2}{(1-\lambda_i)^2} + \frac{2\lambda_i^2}{(1-\sqrt{\lambda_i})^2(1-\lambda_i)}\sum_{t=1}^{T-1}\left\|\beta_t^{(i)}\right\|^2 \\ \leq & \frac{2\left\|y_1^{(i)}\right\|^2}{(1-\lambda)^2} + \frac{2\lambda^2}{(1-\sqrt{\lambda})^2(1-\lambda)}\sum_{t=1}^{T-1}\left\|\beta_t^{(i)}\right\|^2.\end{aligned} \tag{15}$$

Denote $C_1 = \max\left\{\frac{1}{1-|v|^2}, \frac{1}{(1-\lambda)^2}\right\}$ and $C_2 = \max\left\{\frac{\lambda_n^2}{(1-|v|^2)}, \frac{\lambda^2}{(1-\sqrt{\lambda})^2(1-\lambda)}\right\}$. From (14) and (15), we have

$$\sum_{t=1}^{T}\left\|y_t^{(i)}\right\|^2 \leq 2C_1\left\|y_1^{(i)}\right\|^2 + 2C_2\sum_{t=1}^{T-1}\left\|\beta_t^{(i)}\right\|^2. \tag{16}$$

We next bound $\beta_t^{(i)}$

$$
\begin{aligned}
&\mathbb{E}\sum_{i=2}^{n}\|\beta_t^{(i)}\|^2\\
=&\sum_{i=2}^{n}\gamma^2\mathbb{E}\|h_t^{(i)}-h_{t-1}^{(i)}\|^2\\
=&\gamma^2\sum_{i=2}^{n}\mathbb{E}\left\|G\left(X_t;\xi_t\right)P\boldsymbol{e}^{(i)}-G\left(X_{t-1};\xi_{t-1}\right)P\boldsymbol{e}^{(i)}\right\|^2\\
\leq&\gamma^2\sum_{i=1}^{n}\mathbb{E}\left\|G\left(X_t;\xi_t\right)P\boldsymbol{e}^{(i)}-G\left(X_{t-1};\xi_{t-1}\right)P\boldsymbol{e}^{(i)}\right\|^2\\
=&\gamma^2\mathbb{E}\left\|G\left(X_t;\xi_t\right)P-G\left(X_{t-1};\xi_{t-1}\right)P\right\|_F^2\\
=&\gamma^2\mathbb{E}\left\|G\left(X_t;\xi_t\right)-G\left(X_{t-1};\xi_{t-1}\right)\right\|_F^2\quad\text{(due to Lemma 5 )}\\
=&\gamma^2\sum_{i=1}^{n}\mathbb{E}\left\|\nabla F_i\left(\boldsymbol{x}_t^{(i)};\xi_t^{(i)}\right)-\nabla F_i\left(\boldsymbol{x}_{t-1}^{(i)};\xi_{t-1}^{(i)}\right)\right\|^2\\
=&\gamma^2\sum_{i=1}^{n}\mathbb{E}\left\|\left(\nabla F_i\left(\boldsymbol{x}_t^{(i)};\xi_t^{(i)}\right)-\nabla f_i(\boldsymbol{x}_t^{(i)})\right)-\left(F_i\left(\boldsymbol{x}_{t-1}^{(i)};\xi_{t-1}^{(i)}\right)-\nabla f_i(\boldsymbol{x}_{t-1}^{(i)})\right)+\left(\nabla f_i\left(\boldsymbol{x}_t^{(i)}\right)-\nabla f_i\left(\boldsymbol{x}_{t-1}^{(i)}\right)\right)\right\|^2\\
=&3\gamma^2\sum_{i=1}^{n}\mathbb{E}\left\|\nabla F_i\left(\boldsymbol{x}_t^{(i)};\xi_t^{(i)}\right)-\nabla f_i(\boldsymbol{x}_t^{(i)})\right\|^2+3\gamma^2\sum_{i=1}^{n}\left\|F_i\left(\boldsymbol{x}_{t-1}^{(i)};\xi_{t-1}^{(i)}\right)-\nabla f_i(\boldsymbol{x}_{t-1}^{(i)})\right\|^2\\
&+3\gamma^2\sum_{i=1}^{n}\left\|\nabla f_i\left(\boldsymbol{x}_t^{(i)}\right)-\nabla f_i\left(\boldsymbol{x}_{t-1}^{(i)}\right)\right\|^2\\
\leq&6\gamma^2 n\sigma^2+3\gamma^2\sum_{i=1}^{n}\mathbb{E}\left\|\nabla f_i\left(\boldsymbol{x}_t^{(i)}\right)-\nabla f_i\left(\boldsymbol{x}_{t-1}^{(i)}\right)\right\|^2\\
\leq&6\gamma^2 n\sigma^2+3\gamma^2\sum_{i=1}^{n}L^2\mathbb{E}\left\|\boldsymbol{x}_t^{(i)}-\boldsymbol{x}_{t-1}^{(i)}\right\|^2\\
=&6\gamma^2 n\sigma^2+3\gamma^2L^2\sum_{i=1}^{n}\mathbb{E}\left\|Y_tP^\top\boldsymbol{e}^{(i)}-Y_{t-1}P^\top\boldsymbol{e}^{(i)}\right\|^2\\
=&6\gamma^2 n\sigma^2+3\gamma^2L^2\mathbb{E}\left\|Y_tP^\top-Y_{t-1}P^\top\right\|_F^2\\
=&6\gamma^2 n\sigma^2+3\gamma^2L^2\mathbb{E}\left\|Y_t-Y_{t-1}\right\|_F^2\quad\text{(due to Lemma 5 )}\\
=&6\gamma^2 n\sigma^2+3\gamma^2L^2\sum_{i=1}^{n}\mathbb{E}\left\|y_t^{(i)}-y_{t-1}^{(i)}\right\|^2.
\end{aligned}\tag{17}
$$

Combing (16) and (17), we have

$$
\begin{aligned}
\sum_{i=2}^{n}\sum_{t=1}^{T}\|y_t^{(i)}\|^2\leq&2C_1\|Y_1\|_F^2+2C_2\sum_{i=2}^{n}\sum_{t=1}^{T-1}\|\beta_t^{(i)}\|^2\\
\leq&2C_1\|Y_1\|_F^2+2C_2\sum_{t=1}^{T-1}\left(6\gamma^2n\sigma^2+3\gamma^2L^2\sum_{i=1}^{n}\mathbb{E}\left\|y_t^{(i)}-y_{t-1}^{(i)}\right\|^2\right)\\
\leq&2C_1\|Y_1\|_F^2+12C_2\gamma^2n\sigma^2T+6C_2\gamma^2L^2\sum_{i=1}^{n}\sum_{t=1}^{T-1}\mathbb{E}\left\|y_t^{(i)}-y_{t-1}^{(i)}\right\|^2.
\end{aligned}\tag{18}
$$

The next step is to bound $\mathbb{E}\|y_t^{(1)} - y_{t-1}^{(1)}\|^2$. Because

$$y_t^{(1)} = X_t P \boldsymbol{e}^{(1)} = X_t \boldsymbol{v}_1 = X_t \frac{1}{\sqrt{n}} \mathbf{1}_n = \overline{X_t}\sqrt{n},$$

what we need to bound is $\mathbb{E}\|\overline{X}_{t+1} - \overline{X}_t\|^2$. From (4), we have $\overline{X}_{t+1} = \overline{X}_t - \gamma \overline{G}_t$. Therefore

$$\mathbb{E}\left\|\overline{X}_{t+1} - \overline{X}_t\right\|^2 = \gamma^2 \mathbb{E}\left\|\overline{G}_t\right\|^2 = \gamma^2 \mathbb{E}\|\overline{G}_t - \overline{\nabla} f(X_t)\|^2 + \gamma^2 \left\|\overline{\nabla} f(X_t)\right\|^2 \leq \frac{\gamma^2\sigma^2}{n} + \gamma^2 \left\|\overline{\nabla} f(X_t)\right\|^2,$$

and we have the follow bound for $\mathbb{E}\|y_t^{(1)} - y_{t-1}^{(1)}\|^2$:

$$\mathbb{E}\left\|y_{t+1}^{(1)} - y_t^{(1)}\right\|^2 \leq \gamma^2\sigma^2 + n\gamma^2 \left\|\overline{\nabla} f(X_t)\right\|^2. \tag{19}$$

Combing (18) and (19) we get

$$\begin{aligned}
\sum_{i=2}^{n}\sum_{t=1}^{T} \|y_t^{(i)}\|^2 \leq& 2C_1\|Y_1\|_F^2 + 12C_2\gamma^2 n\sigma^2 T + 6C_2\gamma^4L^2\sigma^2 T + 6C_2\gamma^4L^2 n \sum_{t=1}^{T-1}\left\|\overline{\nabla f}(X_t)\right\|^2 \\
&+ 6C_2\gamma^2L^2 \sum_{i=2}^{n}\sum_{t=1}^{T-1} \mathbb{E}\left\|y_t^{(i)} - y_{t-1}^{(i)}\right\|^2 \\
\leq& 2C_1\|Y_1\|_F^2 + 12C_2\gamma^2 n\sigma^2 T + 6C_2\gamma^4L^2\sigma^2 T + 6C_2\gamma^4L^2 n \sum_{t=1}^{T-1}\left\|\overline{\nabla f}(X_t)\right\|^2 \\
&+ 6C_2\gamma^2L^2 \sum_{i=2}^{n}\sum_{t=1}^{T-1} 2\mathbb{E}\left(\left\|y_t^{(i)}\right\|^2 + \left\|y_{t-1}^{(i)}\right\|^2\right) \\
\leq& 2C_1\|Y_1\|_F^2 + 12C_2\gamma^2 n\sigma^2 T + 6C_2\gamma^4L^2\sigma^2 T + 6C_2\gamma^4L^2 n \sum_{t=1}^{T-1}\left\|\overline{\nabla f}(X_t)\right\|^2 \\
&+ 6C_2\gamma^2L^2 \sum_{i=2}^{n}\sum_{t=1}^{T-1} 2\mathbb{E}\left(\left\|y_t^{(i)}\right\|^2 + \left\|y_t^{(i)}\right\|^2\right) \quad \text{(due to } y_0^{(i)} = \mathbf{0}\text{)} \\
\leq& 2C_1\|Y_1\|_F^2 + 12C_2\gamma^2 n\sigma^2 T + 6C_2\gamma^4L^2\sigma^2 T + 6C_2\gamma^4L^2 n \sum_{t=1}^{T-1}\left\|\overline{\nabla f}(X_t)\right\|^2 \\
&+ 24C_2\gamma^2L^2 \sum_{i=2}^{n}\sum_{t=1}^{T-1} \mathbb{E}\left\|y_t^{(i)}\right\|^2, \\
\left(1 - 24C_2\gamma^2L^2\right)\sum_{i=2}^{n}\sum_{t=1}^{T} \|y_t^{(i)}\|^2 \leq& 2C_1\|Y_1\|_F^2 + 12C_2\gamma^2 n\sigma^2 T + 6C_2\gamma^4L^2\sigma^2 T + 6C_2\gamma^4L^2 n \sum_{t=1}^{T-1}\left\|\overline{\nabla f}(X_t)\right\|^2.
\end{aligned}$$

Together with (12) and $X_0 = 0$, we have

$$\begin{aligned}
\left(1 - 24C_2\gamma^2L^2\right)\sum_{i=1}^{n}\sum_{t=1}^{T} \left\|\overline{X}_t - \boldsymbol{x}_t^{(i)}\right\|^2 \leq& 2C_1\|Y_1\|_F^2 + 12C_2\gamma^2 n\sigma^2 T + 6C_2\gamma^4L^2\sigma^2 T + 6C_2\gamma^4L^2 n \sum_{t=1}^{T-1}\left\|\overline{\nabla f}(X_t)\right\|^2 \\
\text{(due to } \|X_1\|_F = \|Y_1\|_F\text{)} \quad \leq& 2C_1\|X_1\|_F^2 + 12C_2\gamma^2 n\sigma^2 T + 6C_2\gamma^4L^2\sigma^2 T + 6C_2\gamma^4L^2 n \sum_{t=1}^{T-1}\left\|\overline{\nabla f}(X_t)\right\|^2.
\end{aligned}$$

Actually, when $\lambda_n \leq -\frac{1}{3}$, we have $|v_n| \geq 1$, then $\|y_t^{(n)}\|^2 \propto t$ and

$$\frac{1}{T}\sum_{i=1}^{n}\sum_{t=1}^{T} \left\|\overline{X}_t - \boldsymbol{x}_t^{(i)}\right\|^2 \leq T.$$

The algorithm would fail to converge in this situation, and this is why $-1/3$ is the infimum of $\lambda_n$. $\square$

**Lemma 8.** *Following the Assumption 1, we have*

$$\mathbb{E}f(\overline{X}_{t+1}) \leq \mathbb{E}f(\overline{X}_t) - \frac{\gamma_t}{2}\mathbb{E}\|\nabla f(\overline{X}_t)\|^2 - \left(\frac{\gamma_t}{2} - \frac{L\gamma_t^2}{2}\right)\mathbb{E}\|\overline{\nabla f}(X_t)\|^2 + \frac{\gamma_t}{2}\mathbb{E}\|\nabla f(\overline{X}_t) - \overline{\nabla f}(X_t)\|^2 + \frac{L\gamma_t^2}{2n}\sigma^2.$$

*Proof.* From (4), we have

$$\overline{X}_{t+1} = \overline{X}_t - \gamma_t \overline{G}(X_t;\xi_t).$$

From item 1 of Assumption 1, we know that $f$ has a $L$-Lipschitz continuous gradient. So, we have

$$\begin{aligned}
\mathbb{E}f(\overline{X}_{t+1}) \leq & \mathbb{E}f(\overline{X}_t) + \mathbb{E}\left\langle \nabla f(\overline{X}_t), -\gamma_t\overline{G}(X_t;\xi_t)\right\rangle + \frac{L}{2}\mathbb{E}\left\|-\gamma_t\overline{G}(X_t;\xi_t)\right\|^2 \\
= & \mathbb{E}f(\overline{X}_t) + \mathbb{E}\langle \nabla f(\overline{X}_t), -\gamma_t\mathbb{E}_{\xi_t}\overline{G}(X_t;\xi_t)\rangle + \frac{L\gamma_t^2}{2}\mathbb{E}\|\overline{G}(X_t;\xi_t)\|^2 \\
= & \mathbb{E}f(\overline{X}_t) - \gamma_t\mathbb{E}\langle \nabla f(\overline{X}_t), \overline{\nabla f}(X_t)\rangle + \frac{L\gamma_t^2}{2}\mathbb{E}\|(\overline{G}(X_t;\xi_t) - \overline{\nabla f}(X_t)) + \overline{\nabla f}(X_t)\|^2 \\
= & \mathbb{E}f(\overline{X}_t) - \gamma_t\mathbb{E}\langle \nabla f(\overline{X}_t), \overline{\nabla f}(X_t)\rangle + \frac{L\gamma_t^2}{2}\mathbb{E}\|\overline{G}(X_t;\xi_t) - \overline{\nabla f}(X_t)\|^2 + \frac{L\gamma_t^2}{2}\mathbb{E}\|\overline{\nabla f}(X_t)\|^2 \\
& + L\gamma_t^2\mathbb{E}\langle \mathbb{E}_{\xi_t}\overline{G}(X_t;\xi_t) - \overline{\nabla f}(X_t), \overline{\nabla f}(X_t)\rangle \\
= & \mathbb{E}f(\overline{X}_t) - \gamma_t\mathbb{E}\langle \nabla f(\overline{X}_t), \overline{\nabla f}(X_t)\rangle + \frac{L\gamma_t^2}{2}\mathbb{E}\|\overline{G}(X_t;\xi_t) - \overline{\nabla f}(X_t)\|^2 + \frac{L\gamma_t^2}{2}\mathbb{E}\|\overline{\nabla f}(X_t)\|^2 \\
= & \mathbb{E}f(\overline{X}_t) - \gamma_t\mathbb{E}\langle \nabla f(\overline{X}_t), \overline{\nabla f}(X_t)\rangle + \frac{L\gamma_t^2}{2n^2}\mathbb{E}\left\|\sum_{i=1}^n\left(\nabla F_i(x_t^{(i)};\xi_t^{(i)}) - \nabla f_i(x_t^{(i)})\right)\right\|^2 \\
& + \frac{L\gamma_t^2}{2}\mathbb{E}\|\overline{\nabla f}(X_t)\|^2 \\
= & \mathbb{E}f(\overline{X}_t) - \gamma_t\mathbb{E}\langle \nabla f(\overline{X}_t), \overline{\nabla f}(X_t)\rangle + \frac{L\gamma_t^2}{2n^2}\sum_{i=1}^n\mathbb{E}\left\|\nabla F_i(x_t^{(i)};\xi_t^{(i)}) - \nabla f_i(x_t^{(i)})\right\|^2 \\
& + \sum_{i\neq i'}^n\mathbb{E}\left\langle \mathbb{E}_{\xi_t}\nabla F_i(x_t^{(i)};\xi_t^{(i)}) - \nabla f_i(x_t^{(i)}), \nabla\mathbb{E}_{\xi_t}F_{i'}(x_t^{(i')};\xi_t^{(i')}) - \nabla f_{i'}(x_t^{(i')})\right\rangle + \frac{L\gamma_t^2}{2}\mathbb{E}\|\overline{\nabla f}(X_t)\|^2 \\
\leq & \mathbb{E}f(\overline{X}_t) - \gamma_t\mathbb{E}\langle \nabla f(\overline{X}_t), \overline{\nabla f}(X_t)\rangle + \frac{L\gamma_t^2}{2n}\sigma^2 + \frac{L\gamma_t^2}{2}\mathbb{E}\|\overline{\nabla f}(X_t)\|^2 \\
= & \mathbb{E}f(\overline{X}_t) - \frac{\gamma_t}{2}\mathbb{E}\|\nabla f(\overline{X}_t)\|^2 - \frac{\gamma_t}{2}\mathbb{E}\|\overline{\nabla f}(X_t)\|^2 + \frac{\gamma_t}{2}\mathbb{E}\|\nabla f(\overline{X}_t) - \overline{\nabla f}(X_t)\|^2 + \frac{L\gamma_t^2}{2}\mathbb{E}\|\overline{\nabla f}(X_t)\|^2 \\
& + \frac{L\gamma_t^2}{2n}\sigma^2 \quad (\text{due to } 2\langle \boldsymbol{a}, \boldsymbol{b}\rangle = \|\boldsymbol{a}\|^2 + \|\boldsymbol{b}\|^2 - \|\boldsymbol{a} - \boldsymbol{b}\|^2) \\
= & \mathbb{E}f(\overline{X}_t) - \frac{\gamma_t}{2}\mathbb{E}\|\nabla f(\overline{X}_t)\|^2 - (\frac{\gamma_t}{2} - \frac{L\gamma_t^2}{2})\mathbb{E}\|\overline{\nabla f}(X_t)\|^2 + \frac{\gamma_t}{2}\mathbb{E}\|\nabla f(\overline{X}_t) - \overline{\nabla f}(X_t)\|^2 + \frac{L\gamma_t^2}{2n}\sigma^2, \quad (20)
\end{aligned}$$

which completes the proof. □

**Proof to Theorem 2**

*Proof.* We first estimate the upper bound for $\mathbb{E}\|\nabla f(\overline{X}_t) - \overline{\nabla f}(X_t)\|^2$:

$$\begin{aligned}
\mathbb{E}\|\nabla f(\overline{X}_t) - \overline{\nabla f}(X_t)\|^2 = & \frac{1}{n^2}\mathbb{E}\left\|\sum_{i=1}^n\left(\nabla f_i(\overline{X}_t) - \nabla f_i(\boldsymbol{x}_t^{(i)})\right)\right\|^2 \\
\leq & \frac{1}{n}\sum_{i=1}^n\mathbb{E}\left\|\nabla f_i(\overline{X}_t) - \nabla f_i(\boldsymbol{x}_t^{(i)})\right\|^2 \\
\leq & \frac{L^2}{n}\mathbb{E}\sum_{i=1}^n\left\|\overline{X}_t - \boldsymbol{x}_t^{(i)}\right\|^2. \quad (21)
\end{aligned}$$

Combining (20) in Lemma 8 and (21) yields

$$
\begin{aligned}
&\frac{\gamma_t}{2}\mathbb{E}\|\nabla f(\overline{X_t})\|^2 + \left(\frac{\gamma_t}{2} - \frac{L\gamma_t^2}{2}\right)\mathbb{E}\|\overline{\nabla f}(X_t)\|^2 \\
\le&\mathbb{E}f(\overline{X}_t) - \mathbb{E}f(\overline{X}_{t+1}) + \frac{\gamma_t}{2}\mathbb{E}\|\nabla f(\overline{X_t}) - \overline{\nabla f}(X_t)\|^2 + \frac{L\gamma_t^2}{2n}\sigma^2 \\
\le&\mathbb{E}f(\overline{X}_t) - \mathbb{E}f(\overline{X}_{t+1}) + \frac{L^2\gamma_t}{2n}\sum_{i=1}^n \|\overline{X}_t - x_t^{(i)}\|^2 + \frac{L\gamma_t^2}{2n}\sigma^2.
\end{aligned}
$$

Setting $\gamma_t = \gamma$, we obtain

$$
\mathbb{E}\|\nabla f(\overline{X_t})\|^2 + (1 - L\gamma)\,\mathbb{E}\|\overline{\nabla f}(X_t)\|^2 \le \frac{2}{\gamma}\left(\mathbb{E}f(\overline{X}_t) - f^* - \left(\mathbb{E}f(\overline{X}_{t+1}) - f^*\right)\right) + \frac{L^2}{n}\sum_{i=1}^n \|\overline{X}_t - x_t^{(i)}\|^2 + \frac{L\gamma}{n}\sigma^2. \tag{22}
$$

From Lemma 7 , we have

$$
\begin{aligned}
\left(1 - 24C_2\gamma^2L^2\right)\sum_{i=1}^n\sum_{t=0}^T \left\|\overline{X}_t - \boldsymbol{x}_t^{(i)}\right\|^2 \le& 2C_1\|X_1\|_F^2 + 12C_2\gamma^2 n\sigma^2 T + 6C_2\gamma^4L^2\sigma^2T \\
&+ 6C_2\gamma^4L^2 n\sum_{t=1}^{T-1}\left\|\overline{\nabla f}(X_t)\right\|^2,
\end{aligned}
$$

If $\gamma$ is not too large that satisfies $1 - 24C_2\gamma^2L^2 > 0$, then denote $C_3 = 1 - 24C_2\gamma^2L^2$, we would have

$$
\sum_{i=1}^n\sum_{t=0}^T \left\|\overline{X}_t - \boldsymbol{x}_t^{(i)}\right\|^2 \le \frac{2C_1}{C_3}\|X_1\|_F^2 + \frac{12C_2\gamma^2 n\sigma^2 T}{C_3} + \frac{6C_2\gamma^4L^2\sigma^2T}{C_3} + \frac{6C_2\gamma^4L^2 n}{C_3}\sum_{t=1}^{T-1}\left\|\overline{\nabla f}(X_t)\right\|^2. \tag{23}
$$

Summarizing both sides of (22) and applying (23) yields

$$
\begin{aligned}
&\sum_{t=0}^{T-1}\left(\mathbb{E}\|\nabla f(\overline{X_t})\|^2 + (1 - L\gamma)\,\mathbb{E}\|\overline{\nabla f}(X_t)\|^2\right) \\
\le&\frac{2\mathbb{E}f(\overline{X}_0) - 2f^*}{\gamma} + \frac{L^2}{n}\sum_{t=0}^T\sum_{i=1}^n \mathbb{E}\left\|\overline{X}_t - x_t^{(i)}\right\|^2 + \frac{LT\gamma}{n}\sigma^2 \\
\le&\frac{2(f(0) - f^*)}{\gamma} + \frac{LT\gamma}{n}\sigma^2 + \frac{2L^2C_1}{nC_3}\|X_1\|_F^2 + \frac{12L^2C_2\gamma^2\sigma^2T}{C_3} + \frac{6L^2C_2\gamma^4L^2\sigma^2T}{nC_3} \\
&+ \frac{6L^2C_2\gamma^4L^2}{C_3}\sum_{t=1}^{T-1}\left\|\overline{\nabla f}(X_t)\right\|^2.
\end{aligned}
$$

It implies

$$
\begin{aligned}
&\sum_{t=0}^{T-1}\left(\mathbb{E}\|\nabla f(\overline{X_t})\|^2 + \left(1 - L\gamma - \frac{6L^2C_2\gamma^4L^2}{C_3}\right)\mathbb{E}\|\overline{\nabla f}(X_t)\|^2\right) \\
\le&\frac{2(f(0) - f^*)}{\gamma} + \frac{LT\gamma}{n}\sigma^2 + \frac{2L^2C_1}{nC_3}\|X_1\|_F^2 + \frac{12L^2C_2\gamma^2 n\sigma^2T}{nC_3} + \frac{6L^2C_2\gamma^4L^2\sigma^2T}{nC_3} \\
=&\frac{2(f(0) - f^*)}{\gamma} + \frac{LT\gamma}{n}\sigma^2 + \frac{2L^2C_1\gamma^2}{nC_3}\|G(0;\xi_0)\|_F^2 + \frac{12L^2C_2\gamma^2\sigma^2T}{C_3} + \frac{6L^2C_2\gamma^4L^2\sigma^2T}{nC_3}.
\end{aligned} \tag{24}
$$

However, $\|G(0;\xi_0)\|_F^2$ can be expanded as:

$$
\begin{aligned}
\|G(0,\xi_0)\|_F^2 =& \sum_{i=1}^n \|(\nabla F_i(\mathbf{0},\xi_1) - \nabla f_i(\mathbf{0})) + (\nabla f_i(\mathbf{0}) - \nabla f(\mathbf{0})) + \nabla f(\mathbf{0})\|^2 \\
\le& 3n\sigma^2 + 3n\zeta_0^2 + 3n\|\nabla f(\mathbf{0})\|^2,
\end{aligned} \tag{25}
$$

where $\zeta_0 = \frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(\mathbf{0}) - \nabla f(\mathbf{0})\|^2$ indicates the difference between different workers' dataset at the start point. Combining (24) and (25), then we have

$$\begin{aligned}
&\sum_{t=0}^{T-1}\left(\mathbb{E}\|\nabla f(\overline{X_t})\|^2 + \left(1 - L\gamma - \frac{6L^2C_2\gamma^4L^2}{C_3}\right)\mathbb{E}\|\overline{\nabla f}(X_t)\|^2\right)\\
\leq &\frac{2(f(0)-f^*)}{\gamma} + \frac{LT\gamma}{n}\sigma^2 + \frac{12L^2C_2\gamma^2\sigma^2T}{C_3} + \frac{6L^2C_2\gamma^4L^2\sigma^2T}{nC_3}\\
&+ \frac{6L^2C_1\gamma^2\sigma^2}{C_3} + \frac{6L^2C_1\gamma^2\zeta_0^2}{C_3} + \frac{6L^2C_1\gamma^2}{C_3}\|\nabla f(\mathbf{0})\|^2.
\end{aligned}$$

Then we have

$$\begin{aligned}
&\left(1 - \frac{6L^2C_1\gamma^2}{C_3}\right)\|\nabla f(\mathbf{0})\|^2 + \sum_{t=1}^{T-1}\left(\mathbb{E}\|\nabla f(\overline{X_t})\|^2 + \left(1 - L\gamma - \frac{6L^2C_2\gamma^4L^2}{C_3}\right)\mathbb{E}\|\overline{\nabla f}(X_t)\|^2\right)\\
\leq &\frac{2(f(0)-f^*)}{\gamma} + \frac{LT\gamma}{n}\sigma^2 + \frac{12L^2C_2\gamma^2\sigma^2T}{C_3} + \frac{6L^2C_2\gamma^4L^2\sigma^2T}{nC_3} + \frac{6L^2C_1\gamma^2\sigma^2}{C_3} + \frac{6L^2C_1\gamma^2\zeta_0^2}{C_3}.
\end{aligned}$$

Denote

$$\begin{aligned}
A_1 =&1 - \frac{6L^2C_1\gamma^2}{C_3}\\
A_2 =&1 - L\gamma - \frac{6L^2C_2\gamma^4L^2}{C_3},
\end{aligned}$$

it becomes

$$\begin{aligned}
&A_1\|\nabla f(\mathbf{0})\|^2 + \sum_{t=1}^{T-1}\left(\mathbb{E}\|\nabla f(\overline{X_t})\|^2 + A_2\mathbb{E}\|\overline{\nabla f}(X_t)\|^2\right)\\
\leq &\frac{2(f(0)-f^*)}{\gamma} + \frac{LT\gamma}{n}\sigma^2 + \frac{12L^2C_2\gamma^2n\sigma^2T}{nC_3} + \frac{6L^2C_2\gamma^4L^2\sigma^2T}{nC_3} + \frac{6L^2C_1\gamma^2\sigma^2}{C_3} + \frac{6L^2C_1\gamma^2\zeta_0^2}{C_3}.
\end{aligned}$$

It completes the proof. □

**Proof to Corollary 3**

*Proof.* From the value of $\gamma$, we obtain

$$C_2\gamma^2L^2 \leq \frac{1}{64}, \quad C_1\gamma^2L^2 \leq \frac{1}{36}.$$

Therefore

$$\begin{aligned}
C_3 =&1 - 24C_2\gamma^2L^2 \geq \frac{1}{2},\\
A_1 =&1 - \frac{6L^2C_1\gamma^2}{C_3} \geq \frac{1}{2},\\
A_2 =&1 - L\gamma - \frac{6L^2C_2\gamma^4L^2}{C_3} > 0,\\
\gamma^2 \leq&\frac{n}{nL^2 + \sigma^2T},\\
\gamma^4 \leq&\frac{n^2}{n^2L^4 + \sigma^4T^2}.
\end{aligned}$$

Then we can remove the $\|\overline{\nabla f}(X_t)\|^2$ and $\|\nabla f(\mathbf{0})\|^2$ on the left hand side of (5) in Theorem 2, and (5) becomes

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\overline{X_t})\|^2 \leq & \frac{4(f(0)-f^*)L(8\sqrt{C_2}+6\sqrt{C_1})}{T} + \frac{4(f(0)-f^*)\sigma}{\sqrt{Tn}} \\ & + \frac{2L\sigma}{\sqrt{Tn}} + \frac{48nL^2C_2\sigma^2}{nL^2+\sigma^2T} + \frac{24L^4n\sigma^2C_2}{n^2L^4+\sigma^4T^2} \\ & + \frac{24nL^2C_1\sigma^2}{T(nL^2+\sigma^2T)} + \frac{24L^2C_1\zeta_0^2}{T(nL^2+\sigma^2T)},\end{aligned}$$

which completes the proof. $\square$