MATCHA: Speeding Up Decentralized SGD via Matching Decomposition Sampling

Jianyu Wang[1], Anit Kumar Sahu[2], Zhouyi Yang[1], Gauri Joshi[1], and Soummya Kar[1]

[1]Carnegie Mellon University
[2]Bosch Center for Artificial Intelligence

Abstract

This paper studies the problem of error-runtime trade-off, typically encountered in decentralized training based on stochastic gradient descent (SGD) using a given network. While a denser (sparser) network topology results in faster (slower) error convergence in terms of iterations, it incurs more (less) communication time/delay per iteration. In this paper, we propose MATCHA, an algorithm that can achieve a win-win in this error-runtime trade-off for any arbitrary network topology. The main idea of MATCHA is to parallelize inter-node communication by decomposing the topology into matchings. To preserve fast error convergence speed, it identifies and communicates more frequently over critical links, and saves communication time by using other links less frequently. Experiments on a suite of datasets and deep neural networks validate the theoretical analyses and demonstrate that MATCHA takes up to 5× less time than vanilla decentralized SGD to reach the same training loss.

1   Introduction

Due to the massive size of training datasets used in state-of-the-art machine learning systems, distributing the data and the computation over a network of worker nodes, i.e., data parallelism has attracted a lot of attention in recent years (Nedić et al., 2018; McMahan et al., 2016). In this paper, we consider a decentralized setting without central coordinators (i.e., parameter servers) where nodes can only exchange parameters or gradients with their neighbors. This scenario is common and useful when performing training in sensor networks, multi-agent systems, as well as federated learning on edge devices.

Error-Runtime Trade-off in Decentralized SGD. Previous works in distributed optimization have extensively studied the error convergence of decentralized SGD in terms of iterations or communication rounds (Nedic & Ozdaglar, 2009; Duchi et al., 2012; Yuan et al., 2016; Zeng & Yin, 2016; Towfic et al., 2016; Jakovetic et al., 2018; Scaman et al., 2018), mostly for (strongly) convex loss functions. Recent works have extended the analyses to smooth and non-convex loss functions and subsequently applied it to distributed deep learning (Lian et al., 2017a; Jiang et al., 2017; Wang & Joshi, 2018b; Assran et al., 2018). *However, all of the aforementioned works only focus on iterations complexity, i.e., the number of iterations required to achieve a target error. They do not explicitly consider or demonstrate how the topology affects the training runtime, that is, wall-clock time required to complete each iteration.* Densely-connected networks, when used appropriately,

give faster error convergence. However, they incur a higher communication delay per iteration, which typically increases with the maximal node degree. Thus, in order to achieve the fastest error-versus-wallclock time convergence, it is imperative to jointly analyze the iteration complexity as well as the runtime per iteration. Only a few previous works explore the second factor from a theoretical perspective, see (Dutta et al., 2018; Gupta et al., 2016) — these works consider the centralized parameter server model.

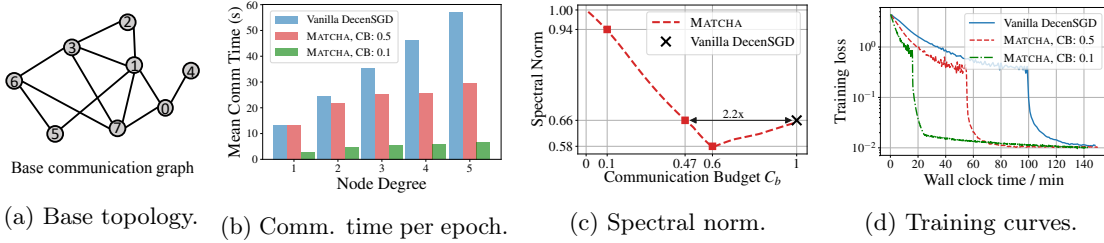(a) Base topology. (b) Comm. time per epoch. (c) Spectral norm. (d) Training curves.

Figure 1: Comparison of vanilla decentralized SGD (DecenSGD) and MATCHA. (a) An example base topology generated using Erdős-Rényi model. (b) MATCHA reduces the communication time non-uniformly. Nodes with higher degrees (for instance node 1 with degree 5) tend to have more redundant links. (c) Lower spectral norm yields better convergence rate. Communication budget C_b represents the average frequency of communication over the links in the network. (d) Loss-versus-time curves when training WideResNet on CIFAR-100.

Main Contributions. To the best of our knowledge, this is the first work that attempts to strike the best error-runtime trade-off in decentralized SGD by carefully *tuning the frequency of inter-node communication*. We propose MATCHA, a decentralized SGD method based on matching decomposition sampling, and demonstrate the effectiveness of MATCHA both theoretically and empirically. The key ideas and main contributions of this paper are as follows.

1. **Saving Communication Time by Using Disjoint Links.** The communication delay of the model synchronization step in decentralized SGD is typically proportional to the maximal node degree (Tsianos et al., 2012). To reduce this communication delay without hurting convergence, we propose a simple, yet powerful idea of decomposing the graph into matchings. Each matching is a set of disjoint links that communicate in parallel, as illustrated by the colored links in Figure 1(a). The probability of activating each matching is optimized so as to maximize the algebraic connectivity of the expected topology (captured by the second smallest eigenvalue λ_2 of the graph Laplacian matrix). This results in more frequent communication over connectivity-critical links (ensuring fast error-versus-iterations convergence) and less frequent over other links (saving inter-node communication time).

2. **Flexible Communication Budget.** MATCHA allows the system designer to set a flexible communication budget C_b, which represents the average frequency of communication over the links in the network. When $C_b = 1$, MATCHA reduces to vanilla decentralized SGD studied in Lian et al. (2017a). When we set $C_b < 1$, MATCHA carefully reduces the communication frequency of each link, depending upon its importance in maintaining the overall connectivity of the graph. For example, observe in Figure 1(b) that by setting $C_b = 0.1$, MATCHA achieves a $1/0.1 = 10\times$ reduction in expected communication time per iteration. Note that the communication reduction is much larger for higher-degree nodes than for low-degree nodes, as shown in Figure 1(b). This judicious

asymmetry in the communication reduction helps MATCHA to preserve fast error-versus-iterations convergence.

3. **Same or Faster Error Convergence than Vanilla Decentralized SGD.** In Section 5 we present a convergence analysis of MATCHA for non-convex objectives and illustrate the dependence of the error on ρ, the spectral norm of the mixing matrix (defined formally later). This analysis shows that for a suitable communication budget, MATCHA achieves the same or smaller ρ as vanilla decentralized SGD — a smaller ρ implies faster error convergence. For example, observe in Figure 1(c) that MATCHA has the same spectral norm as vanilla decentralized SGD (DecenSGD) with a 2.2× less communication budget per iteration, and if we set $C_b = 0.6$ then the spectral norm is even lower. *In this case, contrary to intuition,* MATCHA *not only reduces the per-iteration communication delay but also gives faster error-versus-iterations convergence.*

4. **Experimental Results on Error-versus-wallclock Time Convergence.** In Section 6 we evaluate the performance of MATCHA on a suite of deep learning tasks, including image classification on CIFAR-10/100 and language modeling on Penn Treebank, and for several base graph topologies including Erdős-Rényi and geometric graphs. The empirical results consistently corroborate the theoretical analyses and show that MATCHA can get up to 5.2× reduction in wall-clock time (computation plus communication time) to achieve the same training accuracy as vanilla decentralized SGD, as illustrated in Figure 1(d). Moreover, MATCHA achieves test accuracy that is comparable or better than vanilla decentralized SGD.

5. **Extendable to Other Subgraphs and Computations.** While we currently decompose the topology into matchings, our approach can be extended to other sub-graphs such as edges or cliques. Furthermore, going beyond decentralized SGD, the core idea of MATCHA is extendable to any distributed computation or consensus algorithm that requires frequent synchronization between neighboring nodes.

Connection to Gradient Compression and Quantization Techniques. By reducing the frequency of inter-node communication, MATCHA effectively performs more local SGD updates at each node between model synchronization steps, i.e., consensus steps. Thus, MATCHA belongs to the class of local-update SGD methods recently studied in Stich (2018); Wang & Joshi (2018b,a). An orthogonal way of reducing inter-node communication is to compress or quantize inter-node model updates (Tang et al., 2018; Koloskova et al., 2019). These gradient compression techniques reduce the amount of data that is transmitted per round, whereas local update methods such as MATCHA reduce the frequency of communication. Besides, compression techniques involve encoding and decoding data at each iteration, which incurs additional overhead. MATCHA does not incur such additional overhead at runtime – the sequence of subgraphs is pre-determined before the training starts. MATCHA can be easily combined with these complementary compression techniques to give a further reduction in the overall communication time.

2 Problem Formulation and Preliminaries

Consider a network of m worker nodes. The communication links connecting the nodes are represented by an arbitrary possibly sparse undirected connected graph $\mathcal{G}$ with vertex set $\mathcal{V} = \{1, 2, \ldots, m\}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Each node i can only communicate (i.e., exchange model parameters or gradients) with its neighbors, that is, it can communicate with node j only if $(i, j) \in \mathcal{E}$.

Each worker node i only has access to its own local data distribution $\mathcal{D}_i$. Our objective is to use this network of m nodes to train a model using the joint dataset. In other words, we seek to minimize the objective function $F(\mathbf{x})$, which is defined as follows:

$$F(\mathbf{x}) \triangleq \frac{1}{m} \sum_{i=1}^{m} F_i(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{\mathbf{s} \sim \mathcal{D}_i} \left[\ell(\mathbf{x}; \mathbf{s}) \right] \tag{1}$$

where $\mathbf{x}$ denotes the model parameters (for instance, the weights and biases of a neural network), $F_i(\mathbf{x})$ is the local objective function, $\mathbf{s}$ denotes a single data sample, and $\ell(\mathbf{x}; \mathbf{s})$ is the loss function for sample $\mathbf{s}$, defined by the learning model.

Decentralized SGD (DecenSGD). Decentralized SGD can be tracked back to the seminal work of (Tsitsiklis et al., 1986). It is a natural yet effective way to optimize the empirical risk (1) in the considered decentralized setting. The algorithm simultaneously incorporates the obtained information from the neighborhood (consensus) and the local gradient information as follows[1]:

$$\mathbf{x}_i^{(k+1)} = \underbrace{\sum_{j=1}^{m} W_{ij}}_{\text{consensus step}} \underbrace{\left[\mathbf{x}_j^{(k)} - g(\mathbf{x}_j^{(k)}; \xi_j^{(k)}) \right]}_{\text{local gradient step}} \tag{2}$$

where $\xi_j^{(k)}$ denotes a mini-batch sampled from local data distribution $\mathcal{D}_j$ at iteration k, $g(\mathbf{x}; \xi)$ denotes the stochastic gradient, and W_{ij} is the (i, j)-th element of the mixing matrix $\mathbf{W} \in \mathbb{R}^{m \times m}$. In particular, $W_{ij} \neq 0$ only if node i and node j are connected, i.e., $(i, j) \in \mathcal{E}$. Setting the mixing matrix $\mathbf{W}$ to be symmetric and doubly stochastic is one way to ensure that the nodes reach consensus in terms of converging to the same stationary point. For instance, if node 1 is only connected to nodes 2 and 3, then the first row of $\mathbf{W}$ can be $[1 - 2\alpha, \alpha, \alpha, 0, \ldots, 0]$, where α is constant.

Communication Time Model. In decentralized SGD methods, each node communicates with all of its neighbors, and the delay in performing this local synchronization typically increases with the node degree. Thus, the node with the highest degree in the graph becomes the bottleneck and the communication time per iteration can be modeled as:

$$\text{Comm. Time per Iteration} = t(\Delta(\mathcal{G})), \tag{3}$$

where Δ is the maximal degree of the graph and $t(\cdot)$ is a monotonically increasing function. For brevity of presentation, we will focus on a linear-scaling rule (i.e., $t(\Delta) = \Delta$) in this paper, as considered in previous works (Tsianos et al., 2012; Chow et al., 2016; Lian et al., 2017b) and also observed in experimental result Figure 1(b). But MATCHA can be easily extended to other delay-scaling rule $t(\cdot)$. Without loss of generality, we can assume the communication (sending and receiving model parameters) over one link costs 1 unit of time. Thus, the communication per iteration takes at least $t(\Delta)$ units of time.

Preliminaries from Graph Theory. The communication graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ can be abstracted as an adjacency matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$. In particular, $A_{ij} = 1$ if $(i, j) \in \mathcal{E}$; $A_{ij} = 0$ otherwise. The graph Laplacian $\mathbf{L}$ is defined as $\mathbf{L} = \text{diag}(d_1, \ldots, d_m) - \mathbf{A}$, where d_i denotes the i-th node's degree. When $\mathcal{G}$ is a connected graph, the second smallest eigenvalue λ_2 of the graph Laplacian is strictly greater than 0 and referred to as *algebraic connectivity* in Bollobás (2013). A larger value of λ_2 implies a denser graph. Moreover, we will use the notion of *matching*, defined as follows: a matching in $\mathcal{G}$ is a subgraph of $\mathcal{G}$, in which each vertex is incident with at most one edge.

---

[1]One can also use another update rule: $\mathbf{x}_i^{(k+1)} = \sum_{j=1}^{m} W_{ji} \mathbf{x}_j^{(k)} - g(\mathbf{x}_i^{(k)}; \xi_i^{(k)})$. All insights and conclusions in this paper will remain the same.

3 MATCHA: Proposed Matching Decomposition Sampling Strategy

Following the intuition that it is beneficial to *communicate over critical links more frequently and less over other links*, the algorithm consists of three key steps as follows, all of which can be done before training starts. A brief illustration is also shown in Figure 2.
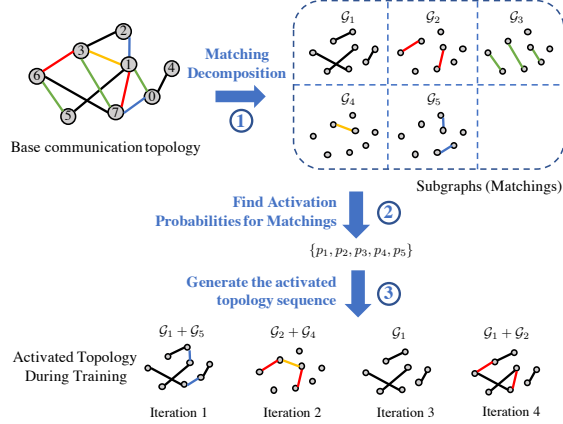

Figure 2: Illustration of MATCHA. Given the base communication graph, we decompose it into disjoint subgraphs (in particular, matchings, in order to allow parallel communications). Then, at each communication round, we carefully sample a subset of these matchings to construct a sparse subgraph of the base topology. Worker nodes are synchronized only through the activated topology.

Step 1: Matching Decomposition. First, we decompose the base communication graph into total M disjoint matchings, i.e., $\mathcal{G}(\mathcal{V}, \mathcal{E}) = \bigcup_{j=1}^{M} \mathcal{G}_j(\mathcal{V}, \mathcal{E}_j)$ and $\mathcal{E}_i \bigcap \mathcal{E}_j = \varnothing, \forall i \neq j$, as illustrated in Figure 2. We can use any matching decomposition algorithm to find the M disjoint matchings. For example, the polynomial-time edge coloring algorithm in Misra & Gries (1992), provably guarantees that the number of matchings M equals to either $\Delta(\mathcal{G})$ or $\Delta(\mathcal{G}) + 1$, where $\Delta(\mathcal{G})$ is the maximal node degree.

For the communication time model where the delay is a monotonically increasing function of the node degree, using matchings is a natural solution because they minimize the effective node degree while maximizing the number of parallel information exchanges between nodes. Since all the nodes in a matching have degree one, the inter-node links are disjoint and these links can operate in parallel. Thus, according to our communication delay model, the communication time for one matching is exactly $t(1) = 1$ unit.

Observe that communicating sequentially over all the matchings is also a simple and efficient way to implement the consensus step in decentralized training. Thus, the communication time per iteration will be equal to the number of matchings M.

Step 2: Computing Matching Activation Probabilities. In order to control the total communication time per iteration, we assign an independent Bernoulli random variable B_j, which is 1 with probability p_j and 0 otherwise, to each matching $\forall j \in \{1, \ldots, M\}$. The links in matching j will be used for information exchange between the corresponding worker nodes only when the

realization of B_j is 1. As a result, the communication time per iteration can be written as

$$\text{Expected Comm. Time} = \mathbb{E}\left[\sum_{j=1}^{M}\mathsf{B}_j\right] = \sum_{j=1}^{M}p_j. \tag{4}$$

where p_j is the *activation probability* of the jth matching. When all p_j's equal to 1, all the M matchings are activated in every iteration. Thus, when $p_j = 1$ for all $j \in \{1, \ldots, M\}$, the algorithm reduces to vanilla DecenSGD and takes M units of time to finish one consensus step. To reduce the expected communication time per iteration given by (4), we define *communication budget* $C_b > 0$, and impose the constraint $\sum_{j=1}^{M}p_j \leq C_b M$. For example, $C_b = 0.1$ means that the expected communication time per iteration of MATCHA be 10% of the time per iteration of vanilla DecenSGD). As mentioned before, the key idea of MATCHA is to give more importance to critical links. This is achieved by choosing a set of activation probabilities that maximize the connectivity of the expected graph given a communication time constraint. That is, we solve the optimization problem:

$$\begin{aligned}
\max_{p_1,\ldots,p_M} \quad & \lambda_2\left(\textstyle\sum_{j=1}^{M}p_j\mathbf{L}_j\right) \\
\text{subject to} \quad & \textstyle\sum_{j=1}^{M}p_j \leq C_b \cdot M, \\
& 0 \leq p_j \leq 1, \; \forall j \in \{1, 2, \ldots, M\},
\end{aligned} \tag{5}$$

where $\mathbf{L}_j$ denotes the Laplacian matrix of the j-th subgraph and $\sum_{j=1}^{M}p_j\mathbf{L}_j$ can be considered as the Laplacian of the expected graph. Moreover, recall that λ_2 represents the algebraic connectivity and is a concave function (Kar & Moura, 2008; Bollobás, 2013). Thus, it follows that (5) is a convex problem and can be solved efficiently. Typically, a larger value of λ_2 implies a better-connected graph.

Step 3: Generating Random Topology Sequence. Given the activation probabilities obtained by solving (5), in each iteration, we generate an independent Bernoulli random variable B_j for each matching. Thus, in the k-th iteration, the activated topology $\mathcal{G}^{(k)} = \bigcup_{j=1}^{M}\mathsf{B}_j^{(k)}\mathcal{G}_j$, which is sparse or even disconnected. Consequently, one can also obtain the corresponding Laplacian matrix sequence: $\mathbf{L}^{(k)} = \sum_{j=1}^{M}\mathsf{B}_j^{(k)}\mathbf{L}_j, \forall k \in \{1, 2, \ldots\}$. *All of these information can be obtained and assigned apriori to worker nodes before starting the training procedure.*

Extension to Other Design Choices. Our proposed MATCHA framework of activating different subgraphs in each iteration is very general – it can be extended to various other delay models, graph decomposition methods and algorithms involving decentralized averaging. For example, instead of using a linear-delay scaling model, one can assume the communication time is a general increasing function $t(\Delta(\mathcal{G}))$. In this case, we only need to change the constraint in (7) to $t(\sum_{j=1}^{M}p_j) \leq C_b \cdot t(M)$. Instead of activating all matchings independently, one can choose to activate only one matching at each iteration. Instead of assuming all links cost same amount of time, one can model the communication time for each link as a random variable and modify the formula (4) accordingly. Finally, instead of decomposing the base topology into matchings, each subgraph can be a single edge or a clique in the base graph.

Why not use Periodic Decentralized SGD? A naive way to reduce the communication time is to perform infrequent synchronization, also called *Periodic DecenSGD (P-DecenSGD)*, and studied in previous works (Tsianos et al., 2012; Wang & Joshi, 2018b). In P-DecenSGD, each node makes several local model updates before synchronizing with its neighbors, that is, all links in the base

topology are activated together ($\mathsf{B}_1 = \cdots = \mathsf{B}_M = 1$) after every few iterations. In this case, C_b equals to the communication frequency of the whole base graph. A drawback of this method is that it treats all links equally. By varying the communication frequency across links depending on how they affect the algebraic connectivity of the expected topology, MATCHA can give the same communication reduction as P-DecenSGD but with better error convergence guarantees. In Sections 5 and 6, we will use P-DecenSGD as a benchmark for comparison.

4   Further Optimizing MATCHA For Decentralized SGD

In order to make the best use of the sequence of activated subgraphs generated by MATCHA when running decentralized SGD with a time-varying topology, we need to optimize the proportions in which the local models are averaged together in the consensus step of Eqn. (2). A common practice is to use an equal weight mixing matrix as (Xiao & Boyd, 2004; Kar & Moura, 2008; Duchi et al., 2012):

$$\mathbf{W}^{(k)} = \mathbf{I} - \alpha \mathbf{L}^{(k)} = \mathbf{I} - \alpha \sum_{j=1}^{M} \mathsf{B}_j^{(k)} \mathbf{L}_j, \tag{6}$$

where $\mathbf{L}^{(k)} = \sum_{j=1}^{M} \mathsf{B}_j^{(k)} \mathbf{L}_j$ denotes the graph Laplacian at the k-th iteration. Each matrix $\mathbf{W}^{(k)}$ is symmetric and doubly stochastic by construction, and the parameter α represents the weight of neighbor's information in the consensus step. To guarantee the convergence of decentralized SGD to a stationary point of smooth and non-convex losses (formally stated in Theorem 2), the *spectral norm* $\rho = \left\| \mathbb{E}[\mathbf{W}^{(k)\top}\mathbf{W}^{(k)}] - \frac{1}{m}\mathbf{1}\mathbf{1}^{\top} \right\|_2$ must be less than 1. In general, a smaller value of ρ leads to a smaller optimization error bound.

In the following theorem, we show that for MATCHA with arbitrary communication budget $C_b > 0$, one can always find a value of α for which the resulting spectral norm $\rho < 1$, which in turn guarantees the convergence of MATCHA to a stationary point.

Theorem 1. *Let $\{\mathbf{L}^{(k)}\}$ denote the sequence of Laplacian matrices generated by MATCHA with arbitrary communication budget $C_b > 0$ for a connected base graph $\mathcal{G}$. If the mixing matrix is defined as $\mathbf{W}^{(k)} = \mathbf{I} - \alpha \mathbf{L}^{(k)}$, then there exists a range of α such that $\rho = \left\| \mathbb{E}[\mathbf{W}^{(k)\top}\mathbf{W}^{(k)}] - \frac{1}{m}\mathbf{1}\mathbf{1}^{\top} \right\|_2 < 1$, which guarantees the convergence of MATCHA to a stationary point.*
The value of α can be obtained by solving the following semi-definite programming problem:

$$\begin{aligned}
\min_{\rho, \alpha, \beta} \quad & \rho, \\
subject\ to \quad & \alpha^2 - \beta \leq 0, \\
& \mathbf{I} - 2\alpha \overline{\mathbf{L}} + \beta[\overline{\mathbf{L}}^2 + 2\widetilde{\mathbf{L}}] - \frac{1}{m}\mathbf{1}\mathbf{1}^{\top} \preceq \rho\mathbf{I}
\end{aligned} \tag{7}$$

where β is an auxiliary variable, $\overline{\mathbf{L}} = \sum_{j=1}^{M} p_j \mathbf{L}_j$ and $\widetilde{\mathbf{L}} = \sum_{j=1}^{M} p_j(1 - p_j)\mathbf{L}_j$.

Similar to the optimization problem (5), the semi-definite programming problem (7) needs to be solved only once at the beginning of training, and this additional computation time is negligible compared to the total training time.

Note that the activation probabilities in MATCHA implicitly influence the spectral norm as well. Ideally, one should jointly optimize p_i's and α via a formulation like (7). However, the resulting

optimization problem is non-convex and cannot be solved efficiently. Therefore, in MATCHA, we separately optimize the p_i's and the parameter α. Optimizing p_i's via (5) can be thought of as minimizing an upper bound of the spectral norm ρ. A more detailed discussion on this is given in Appendix B.

5 Error Convergence Analysis

In this section, we study how the optimization error convergence of MATCHA is affected by the choice of communication budget C_b, and how it compares to the convergence of vanilla DecenSGD. In order to facilitate the analysis, we define the averaged iterate as $\overline{\mathbf{x}}^{(k)} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i^{(k)}$ and the lower bound of the objective function as F_{inf}. Since, we focus on general non-convex loss functions, the quantity of interest is the averaged gradient norm: $\frac{1}{K} \sum_{k=1}^{K} \mathbb{E}[\|\nabla F(\overline{\mathbf{x}}^{(k)})\|^2]$. When it approaches zero, the algorithm converges to a stationary point. The analysis is centered around the following common assumptions.

Assumption 1. *Each local objective function $F_i(\mathbf{x})$ is differentiable and its gradient is L-Lipschitz:* $\|\nabla F_i(\mathbf{x}) - \nabla F_i(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|, \forall i \in \{1, 2, \ldots, m\}$.

Assumption 2. *Stochastic gradients at each worker node are unbiased estimates of the true gradient of the local objectives:* $\mathbb{E}[g(\mathbf{x}_i^{(k)}; \xi_i^{(k)}) | \mathcal{F}^{(k)}] = \nabla F_i(\mathbf{x}_i^{(k)}), \forall i \in \{1, 2, \ldots, m\}$, *where $\mathcal{F}^{(k)}$ denotes the sigma algebra generated by noise in the stochastic gradients and the graph activation probabilities until iteration k.*

Assumption 3. *The variance of stochastic gradients at each worker node is uniformly bounded:* $\mathbb{E}[\|g(\mathbf{x}_i^{(k)}; \xi_i^{(k)}) - \nabla F_i(\mathbf{x}_i^{(k)})\|^2 | \mathcal{F}^{(k)}] \leq \sigma^2, \forall i \in \{1, 2, \ldots, m\}$.

Assumption 4. *The deviation of local objectives' gradients are bounded by a non-negative constant:* $\frac{1}{m} \sum_{i=1}^{m} \|\nabla F_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 \leq \zeta^2$.

We first provide a non-asymptotic convergence guarantee for MATCHA, the proof of which is provided in Appendix C.

Theorem 2 (**Non-asymptotic Convergence of MATCHA**). *Suppose that all local models are initialized with $\overline{\mathbf{x}}^{(1)}$ and $\{\mathbf{W}^{(k)}\}_{k=1}^{K}$ is an i.i.d. mixing matrix sequence generated by MATCHA. Under Assumptions 1 to 4, and if the learning rate satisfies $\eta L \leq \min\{1, (\sqrt{\rho^{-1}} - 1)/4\}$, where ρ is the spectral norm (i.e., largest singular value) of matrix $\mathbb{E}[\mathbf{W}^{(k)\top}\mathbf{W}^{(k)}] - \frac{1}{m}\mathbf{1}\mathbf{1}^{\top}$, then after K iterations,*

$$\frac{1}{K} \sum_{k=1}^{K} \mathbb{E}[\|\nabla F(\overline{\mathbf{x}}^{(k)})\|^2] \leq$$

$$\left(\frac{2[F(\overline{\mathbf{x}}^{(1)}) - F_{inf}]}{\eta K} + \frac{\eta L \sigma^2}{m} \right) \frac{1}{1 - 2D} +$$

$$\frac{2\eta^2 L^2 \rho}{1 - \sqrt{\rho}} \left(\frac{\sigma^2}{1 + \sqrt{\rho}} + \frac{3\zeta^2}{1 - \sqrt{\rho}} \right) \frac{1}{1 - 2D} \tag{8}$$

where $D = 6\eta^2 L^2 \rho / (1 - \sqrt{\rho})^2 < 1/2$. In particular, it is guaranteed that $\rho < 1$ for arbitrary communication budget C_b.

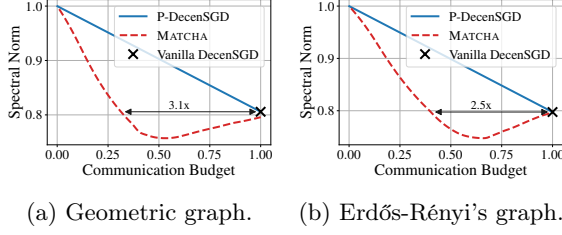(a) Geometric graph. (b) Erdős-Rényi's graph.

Figure 3: Examples on how the spectral norm ρ varies over communication budget in MATCHA. In both (a) and (b), there are 16 worker nodes. MATCHA typically costs $2 - 3\times$ less communication time than vanilla DecenSGD (black crosses) while maintaining the exactly same or even lower value of ρ (i.e., same or better error upper bound).

Consistency with Previous Results. Note that if $\mathbf{W}^{(k)} = \mathbf{1}\mathbf{1}^\top/m$, then $\rho = 0$ and the error bound in Theorem 2 reduces to that for fully synchronous SGD derived in (Bottou et al., 2018). When $\mathbf{W}^{(k)}$ is fixed across iterations, then Theorem 2 reduces to the case of vanilla DecenSGD and recovers the bound in (Lian et al., 2017a; Wang & Joshi, 2018b). Theorem 2 also reveals that the only difference in the optimization error upper bound between MATCHA and vanilla DecenSGD is the value of spectral norm ρ. A smaller value of ρ yields a lower optimization error bound.

Dependence on Communication Budget C_b. While it is difficult to get the analytical form of ρ in terms of the communication budget C_b, in Figure 3, we present some numerical results obtained by solving the optimization problems (5) and (7). Recall that a lower spectral norm ρ means better error-convergence in terms of iterations. Observe that MATCHA takes $2 - 3\times$ less communication time while preserving the same spectral norm as vanilla DecenSGD. By setting a proper communication budget (for instance $C_b \approx 0.5$ in Figure 3a), MATCHA can have even lower spectral norm than vanilla DecenSGD. Besides, to achieve the same spectral norm, MATCHA always requires much less communication budget than periodic DecenSGD. These theoretical findings are corroborated by extensive experiments in Section 6.

Furthermore, if the learning rate is configured properly, MATCHA can achieve a linear speedup in terms of number of worker nodes, matching the same rate as vanilla DecenSGD and fully synchronous SGD.

Corollary 1 (Linear speedup). *Under the same conditions as Theorem 2, if the learning rate is set as $\eta = \sqrt{\frac{m}{K}}$, then after total K iterations, we have*

$$\frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right] = \mathcal{O}\left(\frac{1}{\sqrt{mK}}\right) + \mathcal{O}\left(\frac{m}{K}\right) \tag{9}$$

where all the other constants are subsumed in $\mathcal{O}$.

It is worth noting that when the total iteration K is sufficiently large ($K \geq m^3$), the convergence of MATCHA will be dominated by the first term $1/\sqrt{mK}$.

(a) WideResNet on CIFAR-100. (b) LSTM on Penn Treebank. (c) ResNet on CIFAR-10.

(d) WideResNet on CIFAR-100. (e) LSTM on Penn Treebank. (f) ResNet on CIFAR-10.
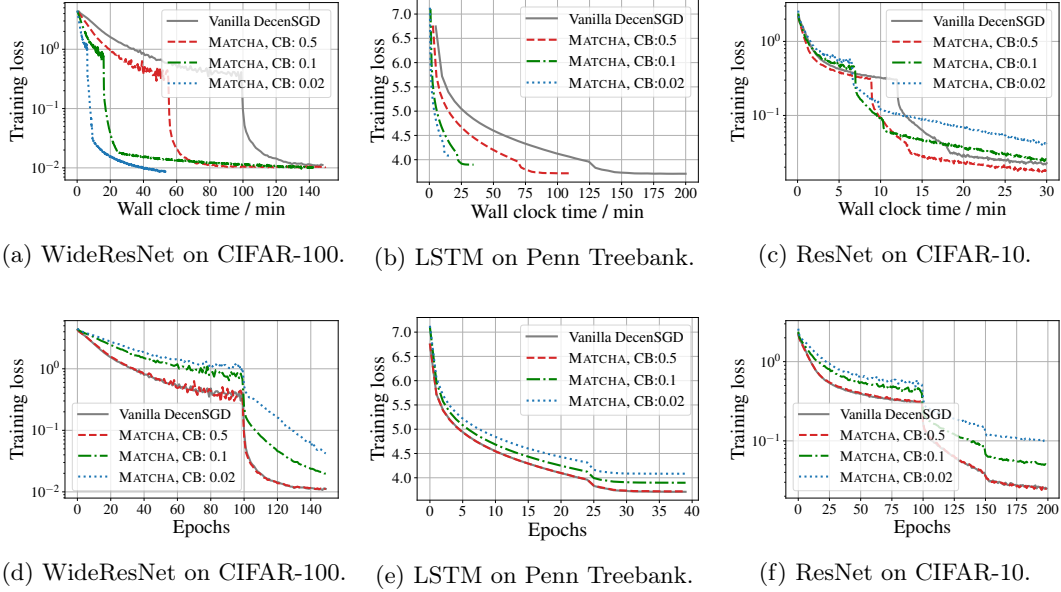
Figure 4: Varying communication budgets C_b in MATCHA. The base communication topology is Figure 1a (an Erdős-Rényi graph with 8 nodes). As predicted by the theoretical result in Figure 1c, when the communication budget is 0.5, MATCHA has nearly the same loss-versus-iteration curves as vanilla DecenSGD but requires only half communication time per iteration. The corresponding test accuracy curves are presented in Appendix A.

6   Experimental Results

We evaluate the performance of MATCHA in multiple deep learning tasks: (1) image classification on CIFAR-10 and CIFAR-100 (Krizhevsky, 2009); (2) Language modeling on Penn Treebank corpus (PTB) dataset (Marcus et al., 1993). All training datasets are evenly partitioned over a network of workers. All algorithms are trained for sufficiently long time until convergence or overfitting. The learning rate is fine-tuned for vanilla DecenSGD and then used for all other algorithms, since we treat MATCHA as an in-place replacement of vanilla DecenSGD. Note that this will in fact disadvantage MATCHA in the comparison with vanilla DecenSGD. A detailed description of the training configurations is provided in Appendix A.1.

Effectiveness of MATCHA. In Figure 4, we evaluate the performance of MATCHA with various communication budgets ($2\%, 10\%, 50\%$ of vanilla DecenSGD). The base communication topology is shown in Figure 2. From Figures 4d to 4f, one can observe that when the communication budget is set to 0.5 (reducing expected communication time per iteration by 50%), MATCHA has the nearly identical training losses as vanilla DecenSGD at every epoch. This empirical finding reinforces the claim in Section 5 regarding the similarity of the algorithms' performance in terms of iterations. When we continue to decrease the communication budget, MATCHA attains significantly faster convergence with respect to wall-clock time in communication-intensive tasks. In particular, on CIFAR-100 (see Figure 4a), MATCHA with $C_b = 0.02$ can take $5\times$ less wall-clock time than vanilla DecenSGD to reach a training loss of 0.1.

(a) Maximal degree is 5.   (b) Maximal degree is 10.   (c) Maximal degree is 13.

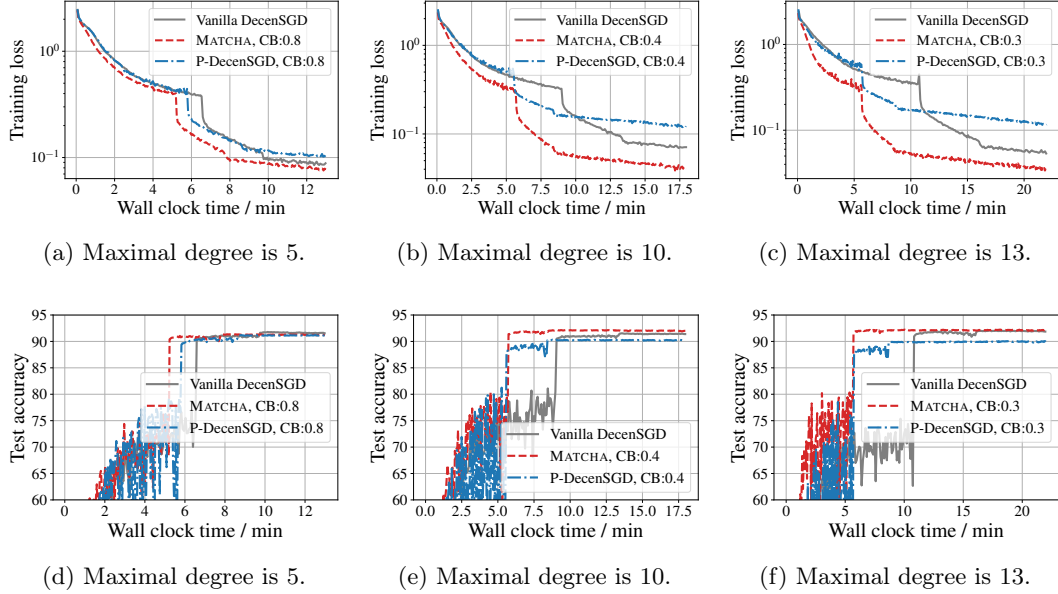(d) Maximal degree is 5.   (e) Maximal degree is 10.   (f) Maximal degree is 13.

Figure 5: Performance comparison on three random-generated geometric graphs with 16 nodes and different levels of connectivity. We train a ResNet-50 on CIFAR-10 dataset. The corresponding base typologies can be found in Appendix A. In particular, the base topology used in Figures 5b and 5e is the same as Figure 3a, which suggests that MATCHA can achieve lower spectral norm than vanilla DecenSGD when $C_b \geq 0.3$. As a consequence, Figures 5b and 5e show that MATCHA can reach a lower training loss and a higher test accuracy than vanilla DecenSGD.

Effects of Base Communication Topology. To further verify the applicability of MATCHA to arbitrary graphs, we evaluate it on different topologies with varying levels of connectivity. In Figure 5, we present experimental results on three random geometric graph topologies that have different maximal degrees. In particular, when the maximal degree is 10 (see Figure 5b), MATCHA with communication budget $C_b = 0.4$ not only reduces the mean communication time per iteration by $1/0.4 = 2.5\times$ but also has lower error than vanilla DecenSGD. This result corroborates the corresponding spectral norm versus communication budget curve shown in Figure 3a. When we further increase the density of the base topology (see Figure 5c), MATCHA can reduce communication time per iteration by $1/0.3 \simeq 3.3\times$ without hurting the error-convergence.

Another interesting observation is that MATCHA gives a more drastic communication reduction for denser base graphs. As shown in Figure 5, along with the increase in the density of the base graph, the training time of vanilla DecenSGD also increases from 13 to 22 minutes to finish 200 epochs. However, in MATCHA, since the effective maximal degree in all cases is maintained to be about 4 by controlling communication budget, the total training time of 200 epochs remains nearly the same (about 11 minutes). Moreover, MATCHA also takes less and less time to achieve a training loss of 0.1, on the contrast to vanilla DecenSGD and P-DecenSGD.

Comparison to Periodic DecenSGD. As discussed in Section 3, a naive way to reduce the communication time per iteration is to use the whole base graph for synchronization after every few iterations (Wang & Joshi, 2018b; Tsianos et al., 2012). Instead, in MATCHA, we allow different

matchings to have different communication frequencies. Similar to the theoretical simulations in Figure 3, the results in Figure 5 show that given a fixed communication budget C_b, MATCHA consistently outperforms periodic DecenSGD. More results are presented in the Appendix.

7   Concluding Remarks

In this paper, we propose MATCHA to reduce the communication delay of decentralized SGD over an arbitrary node topology. The key idea in MATCHA is that workers communicate over the connectivity-critical links with higher probability, which is achieved via matching decomposition sampling. Rigorous theoretical analysis and experimental results show that MATCHA can reduce the communication delay while maintaining the same (or even improve) error-versus-iterations convergence rate. Future directions include adaptively changing the communication budget similar to (Wang & Joshi, 2018a), extending MATCHA to directed communication graphs (Assran et al., 2018) and other decentralized computing algorithms.

References

Assran, Mahmoud, Loizou, Nicolas, Ballas, Nicolas, and Rabbat, Michael. Stochastic gradient push for distributed deep learning. *arXiv preprint arXiv:1811.10792*, 2018.

Bollobás, Béla. *Modern graph theory*, volume 184. Springer Science & Business Media, 2013.

Bottou, Léon, Curtis, Frank E, and Nocedal, Jorge. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

Chow, Yat-Tin, Shi, Wei, Wu, Tianyu, and Yin, Wotao. Expander graph and communication-efficient decentralized optimization. In *2016 50th Asilomar Conference on Signals, Systems and Computers*, pp. 1715–1720. IEEE, 2016.

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

Dutta, Sanghamitra, Joshi, Gauri, Ghosh, Soumyadip, Dube, Parijat, and Nagpurkar, Priya. Slow and stale gradients can win the race: Error-runtime trade-offs in distributed SGD. *arXiv preprint arXiv:1803.01113*, 2018.

Gupta, Suyog, Zhang, Wei, and Wang, Fei. Model accuracy and runtime tradeoff in distributed deep learning: A systematic study. In *IEEE 16th International Conference on Data Mining (ICDM)*, pp. 171–180. IEEE, 2016.

Jakovetic, Dusan, Bajovic, Dragana, Sahu, Anit Kumar, and Kar, Soummya. Convergence rates for distributed stochastic optimization over random networks. In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 4238–4245. IEEE, 2018.

Jiang, Zhanhong, Balu, Aditya, Hegde, Chinmay, and Sarkar, Soumik. Collaborative deep learning in fixed topology networks. In *Advances in Neural Information Processing Systems*, pp. 5906–5916, 2017.

Kar, Soummya and Moura, José MF. Sensor networks with random links: Topology design for distributed consensus. *IEEE Transactions on Signal Processing*, 56(7):3315–3326, 2008.

Koloskova, Anastasia, Stich, Sebastian U, and Jaggi, Martin. Decentralized stochastic optimization and gossip algorithms with compressed communication. *arXiv preprint arXiv:1902.00340*, 2019.

Krizhevsky, Alex. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Lian, Xiangru, Zhang, Ce, Zhang, Huan, Hsieh, Cho-Jui, Zhang, Wei, and Liu, Ji. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pp. 5336–5346, 2017a.

Lian, Xiangru, Zhang, Wei, Zhang, Ce, and Liu, Ji. Asynchronous decentralized parallel stochastic gradient descent. *arXiv preprint arXiv:1710.06952*, 2017b.

Marcus, Mitchell, Santorini, Beatrice, and Marcinkiewicz, Mary Ann. Building a large annotated corpus of english: The penn treebank. 1993.

McMahan, H Brendan, Moore, Eider, Ramage, Daniel, Hampson, Seth, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

Misra, Jayadev and Gries, David. A constructive proof of vizing's theorem. In *Information Processing Letters*. Citeseer, 1992.

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

Nedić, Angelia, Olshevsky, Alex, and Rabbat, Michael G. Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106(5):953–976, 2018.

Press, Ofir and Wolf, Lior. Using the output embedding to improve language models. *arXiv preprint arXiv:1608.05859*, 2016.

Scaman, Kevin, Bach, Francis, Bubeck, Sébastien, Massoulié, Laurent, and Lee, Yin Tat. Optimal algorithms for non-smooth distributed optimization in networks. In *Advances in Neural Information Processing Systems*, pp. 2740–2749, 2018.

Stich, Sebastian U. Local SGD converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

Tang, Hanlin, Gan, Shaoduo, Zhang, Ce, Zhang, Tong, and Liu, Ji. Communication compression for decentralized training. In *Advances in Neural Information Processing Systems*, pp. 7652–7662, 2018.

Towfic, Zaid J, Chen, Jianshu, and Sayed, Ali H. Excess-risk of distributed stochastic learners. *IEEE Transactions on Information Theory*, 62(10):5753–5785, 2016.

Tsianos, Konstantinos, Lawlor, Sean, and Rabbat, Michael G. Communication/computation tradeoffs in consensus-based distributed optimization. In *Advances in neural information processing systems*, pp. 1943–1951, 2012.

Tsitsiklis, John, Bertsekas, Dimitri, and Athans, Michael. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9): 803–812, 1986.

Wang, Jianyu and Joshi, Gauri. Adaptive communication strategies to achieve the best error-runtime trade-off in local-update SGD. *CoRR*, abs/1810.08313, 2018a.

Wang, Jianyu and Joshi, Gauri. Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms. *arXiv preprint arXiv:1808.07576*, 2018b.

Xiao, Lin and Boyd, Stephen. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

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

Zeng, Jinshan and Yin, Wotao. On nonconvex decentralized gradient descent. *arXiv preprint arXiv:1608.05766*, 2016.

A More Experimental Results

A.1 Detailed Experimental Setting

Image Classification Tasks. CIFAR-10 and CIFAR-100 consist of $60,000$ color images in 10 and 100 classes, respectively. For CIFAR-10 and CIFAR-100 training, we set the initial learning rate as 0.8 and it decays by 10 after 100 and 150 epochs. The mini-batch size per worker node is 64. We train vanilla DecenSGD for 200 epochs and all other algorithms for the same wall-clock time as vanilla DecenSGD.

Language Model Task. The PTB dataset contains $923,000$ training words. A two-layer LSTM with 1500 hidden nodes in each layer Press & Wolf (2016) is adopted. We set the initial learning rate as 40 and it decays by 4 when the training procedure saturates. The mini-batch size per worker node is 10. The embedding size is 1500. All algorithms are trained for 40 epochs.

Machines. The training procedure is performed in a network of nodes, each of which is equipped with one NVIDIA TitanX Maxwell GPU and has a 40 Gbps (5000 MB/s) Ethernet interface. MATCHA is implemented with PyTorch and MPI4Py.
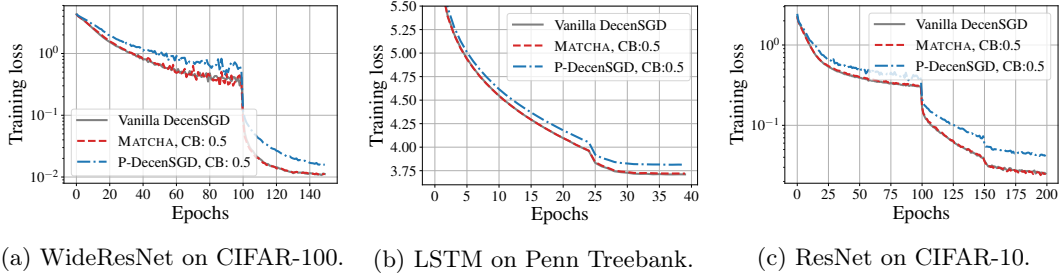
A.2 More Results

(a) WideResNet on CIFAR-100. (b) LSTM on Penn Treebank. (c) ResNet on CIFAR-10.

Figure 6: Comparision of MATCHA and P-DecenSGD. The base communication topology is given in Figure 1(a). While MATCHA has nearly identical error-convergence to vanilla DecenSGD, P-DecenSGD performs consistently worse in all tasks. Note that MATCHA and P-DecenSGD have the same average communication time per iteration.

B Proof of Theorem 1

The proof contains three parts: (1) we first show that the expected activated topology $\sum_{j=1}^{M} p_j \mathbf{L}_j$ is connected, i.e., $\lambda_2(\sum_{j=1}^{M} p_j \mathbf{L}_j) > 0$; (2) then we will prove that if the expected topology is connected, then there must exist an α such that $\rho < 1$ for any arbitrary communication budget; (3) finally, we will show that α can be obtained via solving a semi-definite programming problem.

Recall that $\{p_j\}_{j=1}^{M}$ is the solution of convex optimization problem (5). Let $p_0 = C_b$, then we have

$$\lambda_2(\sum_{j=1}^{M} p_j \mathbf{L}_j) \geq \lambda_2(p_0 \sum_{j=1}^{M} \mathbf{L}_j) = p_0 \lambda_2(\sum_{j=1}^{M} \mathbf{L}_j) > 0.$$

(a) WideResNet on CIFAR-100. (b) LSTM on Penn Treebank. (c) ResNet on CIFAR-10.
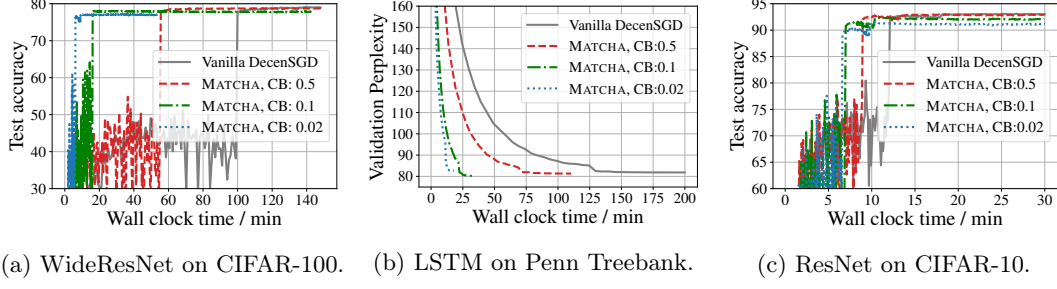
Figure 7: Test accuracy of MATCHA on different training tasks, corresponding to the training loss curves in Figure 4. The base communication topology is given in Figure 1(a). As predicted by the theoretical result Figure 1(c), MATCHA with $C_b = 0.5$ consistently reaches the same test accuracy as vanilla DecenSGD. When we further reduce the communication budget, the final test accuracy of MATCHA only slightly degrades.
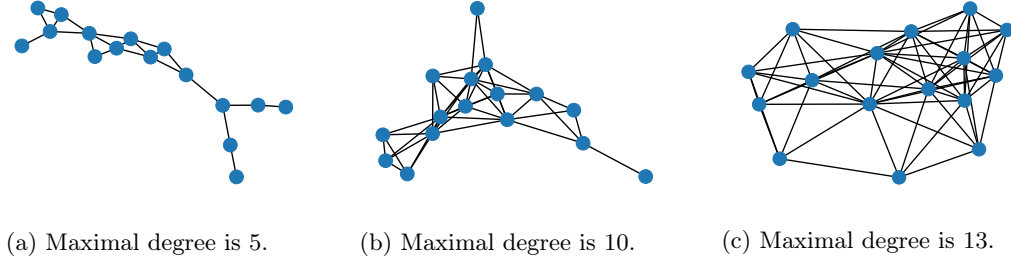

(a) Maximal degree is 5. (b) Maximal degree is 10. (c) Maximal degree is 13.

Figure 8: Different geometric topologies used in Figures 5.

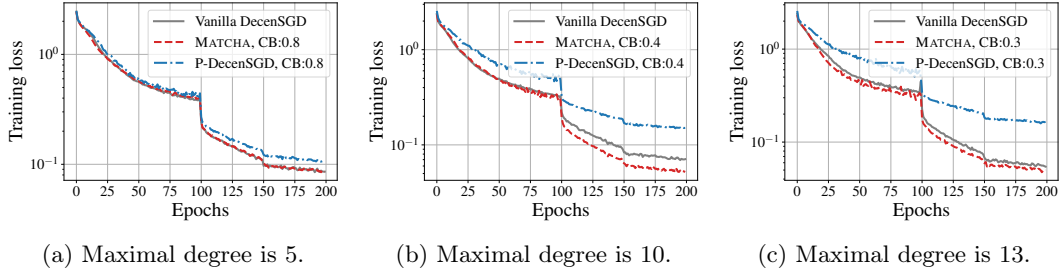(a) Maximal degree is 5. (b) Maximal degree is 10. (c) Maximal degree is 13.

Figure 9: Training loss versus epochs of MATCHA on different topologies with 16 nodes. MATCHA can even have lower training loss than vanilla DecenSGD by setting a proper communication budget.

The last inequality comes from the fact: the base communication topology is connected, i.e., $\lambda_2(\sum_{j=1}^{M} \mathbf{L}_j) > 0$. Here we complete the first part of the proof. Then, recall the definition of ρ and

$\mathbf{W}^{(k)}$, we obtain

$$\left\| \mathbb{E}\left[\mathbf{W}^{(k)\top}\mathbf{W}^{(k)} \right] - \mathbf{J} \right\|_2 = \left\| \mathbb{E}\left[\left(\mathbf{I} - \alpha\mathbf{L}^{(k)} \right)^\top \left(\mathbf{I} - \alpha\mathbf{L}^{(k)} \right) \right] - \mathbf{J} \right\|_2 \tag{10}$$

$$= \left\| \mathbf{I} - 2\alpha\mathbb{E}\left[\mathbf{L}^{(k)} \right] + \alpha^2\mathbb{E}\left[\mathbf{L}^{(k)\top}\mathbf{L}^{(k)} \right] - \mathbf{J} \right\|_2 \tag{11}$$

where $\mathbf{L}^{(k)} = \sum_{j=1}^M \mathsf{B}_j^{(k)}\mathbf{L}_j$. Since $\mathsf{B}_j^{(k)}$'s are i.i.d. across all subgraphs and iterations,

$$\mathbb{E}\left[\mathbf{L}^{(k)} \right] = \sum_{j=1}^M p_j\mathbf{L}_j \tag{12}$$

$$\mathbb{E}\left[\mathbf{L}^{(k)\top}\mathbf{L}^{(k)} \right] = \sum_{j=1}^M p_j^2\mathbf{L}_j^2 + \sum_{j=1}^M\sum_{t=1,t\neq j}^M p_j p_t\mathbf{L}_j^\top\mathbf{L}_t + \sum_{j=1}^M p_j(1-p_j)\mathbf{L}_j^2 \tag{13}$$

$$= \left(\sum_{j=1}^M p_j\mathbf{L}_j \right)^2 + \sum_{j=1}^M p_j(1-p_j)\mathbf{L}_j^2 \tag{14}$$

$$= \left(\sum_{j=1}^M p_j\mathbf{L}_j \right)^2 + 2\sum_{j=1}^M p_j(1-p_j)\mathbf{L}_j. \tag{15}$$

Plugging (12) and (15) back into (11), we get

$$\left\| \mathbb{E}\left[\mathbf{W}^{(k)\top}\mathbf{W}^{(k)} \right] - \mathbf{J} \right\|_2 = \left\| \left(\mathbf{I} - \alpha\sum_{j=1}^M p_j\mathbf{L}_j \right)^2 + 2\alpha^2\sum_{j=1}^M p_j(1-p_j)\mathbf{L}_j - \mathbf{J} \right\|_2$$

$$\leq \left\| \left(\mathbf{I} - \alpha\sum_{j=1}^M p_j\mathbf{L}_j \right)^2 - \mathbf{J} \right\|_2 + 2\alpha^2\left\| \sum_{j=1}^M p_j(1-p_j)\mathbf{L}_j \right\|_2$$

$$= \max\{|1-\alpha\lambda_2|^2, |1-\alpha\lambda_m|^2\} + 2\alpha^2\zeta$$

where λ_i denotes the i-th smallest eigenvalue of matrix $\sum_{j=1}^M p_j\mathbf{L}_j$ and $\zeta \geq 0$ denotes the spectral norm of matrix $\sum_{j=1}^M p_j(1-p_j)\mathbf{L}_j$. Suppose $h_\lambda(\alpha) = (1-\alpha\lambda)^2 + 2\alpha^2\zeta$. Then, we have

$$\frac{\partial h}{\partial\alpha} = -2\lambda(1-\alpha\lambda) + 4\alpha\zeta,$$

$$\frac{\partial^2 h}{\partial\alpha^2} = 2\lambda^2 + 4\zeta > 0.$$

Therefore, $h_\lambda(\alpha)$ is a convex funtion. By setting its derivative to zero, we can get the minimal value:

$$\alpha^* = \frac{\lambda}{\lambda^2 + 2\zeta},$$

$$h_\lambda(\alpha^*) = \frac{4\zeta^2}{(\lambda^2 + 2\zeta)^2} + \frac{2\lambda^2\zeta}{(\lambda^2 + 2\zeta)^2} = \frac{2\zeta}{\lambda^2 + 2\zeta}.$$

We already prove that $\lambda_m \geq \lambda_2 > 0$. Therefore, $\alpha^* > 0$ and $h_\lambda(\alpha^*) < 1$. Furthermore, note that $h_\lambda(0) = 1$ and $h_\lambda(\alpha)$ is a quadratic function. We can conclude that when $\alpha \in (0, 2\alpha^*)$, $h_\lambda(\alpha^*) \leq h_\lambda(\alpha) < 1$. Thus, when $\alpha \in (0, \min\{\frac{2\lambda_2}{\lambda_2^2 + 2\zeta}, \frac{2\lambda_m}{\lambda_m^2 + 2\zeta}\})$, we have

$$\left\| \mathbb{E}\left[\mathbf{W}^{(k)\top}\mathbf{W}^{(k)}\right] - \mathbf{J} \right\|_2 \leq \max\{h_{\lambda_2}(\alpha), h_{\lambda_m}(\alpha)\} < 1.$$

B.1 Formulation of the Semi-Definite Programming Problem

It has been shown that the spectral norm ρ can be expanded as

$$\left\| \mathbb{E}\left[\mathbf{W}^{(k)\top}\mathbf{W}^{(k)}\right] - \mathbf{J} \right\|_2 = \left\| \mathbf{I} - 2\alpha \sum_{j=1}^{M} p_j \mathbf{L}_j + \alpha^2 \left(\sum_{j=1}^{M} p_j \mathbf{L}_j\right)^2 + 2\alpha^2 \sum_{j=1}^{M} p_j(1-p_j)\mathbf{L}_j - \mathbf{J} \right\|_2$$

$$= \left\| \mathbf{I} - 2\alpha\overline{\mathbf{L}} + \alpha^2\overline{\mathbf{L}}^2 + 2\alpha^2\widetilde{\mathbf{L}} - \mathbf{J} \right\|_2$$

where $\overline{\mathbf{L}} = \sum_{j=1}^{M} \mathbf{L}_j$ and $\widetilde{\mathbf{L}} = \sum_{j=1}^{M} p_j(1-p_j)\mathbf{L}_j$. Our goal is to find a value of α that minize the spectral norm:

$$\min_{\alpha} \quad \left\| \mathbf{I} - 2\alpha\overline{\mathbf{L}} + \alpha^2[\overline{\mathbf{L}}^2 + 2\widetilde{\mathbf{L}}] - \mathbf{J} \right\|_2$$

which is equivalent to

$$\begin{aligned} \min_{\rho, \alpha} \quad & \rho \\ \text{subject to} \quad & \mathbf{I} - 2\alpha\overline{\mathbf{L}} + \alpha^2[\overline{\mathbf{L}}^2 + 2\widetilde{\mathbf{L}}] - \frac{1}{m}\mathbf{1}\mathbf{1}^\top \preceq \rho\mathbf{I}. \end{aligned} \tag{16}$$

However, directly solving (16) is N-P hard as it has bilinear matrix inequality constraint. We relax the above optimization problem by introducing an auxiliary variable β as follows:

$$\begin{aligned} \min_{\rho, \alpha, \beta} \quad & \rho \\ \text{subject to} \quad & \alpha^2 - \beta \leq 0, \ \mathbf{I} - 2\alpha\overline{\mathbf{L}} + \beta[\overline{\mathbf{L}}^2 + 2\widetilde{\mathbf{L}}] - \frac{1}{m}\mathbf{1}\mathbf{1}^\top \preceq \rho\mathbf{I}. \end{aligned} \tag{17}$$

Now the constraints become linear matrix inequality constraints and (17) is the standard form of semi-definite programming. However, we need to further show that the solution of (17) is same as (16). We will prove this by contradiction. Suppose $\alpha_+, \beta_+, \rho_+$ are the solution of problem (17) and they satisfy $\alpha_+^2 < \beta_+$. Without loss of generality, we can simply assume $\beta_+ = \alpha_+^2 + c$, where c is a positive constant. Then, we have

$$\mathbf{I} - 2\alpha_+\overline{\mathbf{L}} + (\alpha_+^2 + c)[\overline{\mathbf{L}}^2 + 2\widetilde{\mathbf{L}}] - \frac{1}{m}\mathbf{1}\mathbf{1}^\top \preceq \rho_+\mathbf{I}.$$

Furthermore, according to the definitions of $\overline{\mathbf{L}}$ and $\widetilde{\mathbf{L}}$, both of these matrix are positive semi-definite and have positive largest eigenvalues. As a result, we can obtain

$$\mathbf{I} - 2\alpha_+\overline{\mathbf{L}} + \alpha_+^2[\overline{\mathbf{L}}^2 + 2\widetilde{\mathbf{L}}] - \frac{1}{m}\mathbf{1}\mathbf{1}^\top \prec \mathbf{I} - 2\alpha_+\overline{\mathbf{L}} + (\alpha_+^2 + c)[\overline{\mathbf{L}}^2 + 2\widetilde{\mathbf{L}}] - \frac{1}{m}\mathbf{1}\mathbf{1}^\top \preceq \rho_+\mathbf{I}.$$

That is to say, there must exist ρ_* such that

$$\mathbf{I} - 2\alpha_+ \overline{\mathbf{L}} + \alpha_+^2 [\overline{\mathbf{L}}^2 + 2\widetilde{\mathbf{L}}] - \frac{1}{m} \mathbf{1}\mathbf{1}^\top \preceq \rho_* \mathbf{I} \prec \rho_+ \mathbf{I}.$$

So ρ_+ is not the optimal solution. Our assumptions cannot hold. The solutions of (17) must satisfy $\alpha^2 = \beta$.

B.2 Discussions on the Upper Bound of Spectral Norm

Here, we are going to show that optimizing p_i's via (5) is equivalent to minimizing an upper bound of the spectral norm ρ. Recall that

$$\begin{aligned}
\left\| \mathbb{E}\left[\mathbf{W}^{(k)\top} \mathbf{W}^{(k)} \right] - \mathbf{J} \right\|_2 &= \left\| \mathbf{I} - 2\alpha\overline{\mathbf{L}} + \alpha^2 \overline{\mathbf{L}}^2 + 2\alpha^2 \widetilde{\mathbf{L}} - \mathbf{J} \right\|_2 \\
&\leq \left\| \mathbf{I} - 2\alpha\overline{\mathbf{L}} - \mathbf{J} \right\|_2 + \alpha^2 \left\| \overline{\mathbf{L}}^2 \right\|_2 + 2\alpha^2 \left\| \widetilde{\mathbf{L}} \right\|_2 \\
&= (1 - 2\alpha\lambda_2(\overline{\mathbf{L}})) + \alpha^2 \left\| \overline{\mathbf{L}}^2 \right\|_2 + 2\alpha^2 \left\| \widetilde{\mathbf{L}} \right\|_2.
\end{aligned}$$

According to the definition of $\overline{\mathbf{L}}$ and $\widetilde{\mathbf{L}}$ and repeatedly using Cauchy-Schwartz inequality and triangle inequality, we have

$$\left\| \overline{\mathbf{L}}^2 \right\|_2 \leq \left\| \overline{\mathbf{L}} \right\|_2^2 \leq \left\| \sum_{j=1}^M p_j \mathbf{L}_j \right\|_2^2 \leq \left(\sum_{j=1}^M p_j \left\| \mathbf{L}_j \right\|_2 \right)^2 \leq 4\left(\sum_{j=1}^M p_j \right)^2 \leq 4C_b^2,$$

$$\left\| \widetilde{\mathbf{L}} \right\|_2 = \left\| \sum_{j=1}^M p_j(1 - p_j) \mathbf{L}_j \right\|_2 \leq \sum_{j=1}^M p_j(1 - p_j) \left\| \mathbf{L}_j \right\|_2 \leq 2 \sum_{j=1}^M p_j(1 - p_j) \leq 2 \sum_{j=1}^M p_j \leq 2C_b.$$

As a result, one can get

$$\left\| \mathbb{E}\left[\mathbf{W}^{(k)\top} \mathbf{W}^{(k)} \right] - \mathbf{J} \right\|_2 \leq 1 - 2\alpha\lambda_2(\overline{\mathbf{L}}) + 4\alpha^2 C_b^2 + 4\alpha^2 C_b. \tag{18}$$

Therefore, maximizing $\lambda_2(\overline{\mathbf{L}})$ is equivalent to minimizing the RHS in (18) — an upper bound of the spectral norm ρ.

C Proofs of Theorem 2 and Corollary 1

C.1 Preliminaries

In the proof, we will use the following matrix forms:

$$\begin{aligned}
\mathbf{X}^{(k)} &= \left[\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, \dots, \mathbf{x}_m^{(k)} \right], \\
\mathbf{G}^{(k)} &= \left[g_1(\mathbf{x}_1^{(k)}), g_2(\mathbf{x}_2^{(k)}), \dots, g_m(\mathbf{x}_m^{(k)}) \right], \\
\nabla \mathbf{F}^{(k)} &= \left[\nabla F_1(\mathbf{x}_1^{(k)}), \nabla F_2(\mathbf{x}_2^{(k)}), \dots, \nabla F_m(\mathbf{x}_m^{(k)}) \right].
\end{aligned}$$

Recall the assumptions we make:

$$\|\nabla F_i(\mathbf{x}) - \nabla F_i(\mathbf{y})\| \le L \|\mathbf{x} - \mathbf{y}\|,$$
$$\mathbb{E}\left[g_i(\mathbf{x})|\mathbf{x}\right] = \nabla F_i(\mathbf{x}),$$
$$\mathbb{E}\left[\|g_i(\mathbf{x}) - \nabla F_i(\mathbf{x})\|^2 |\mathbf{x}\right] \le \sigma^2,$$
$$\frac{1}{m}\sum_{i=1}^{m}\|\nabla F_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 \le \zeta^2.$$

The matrix form update rule can be written as:

$$\mathbf{X}^{(k+1)} = \left(\mathbf{X}^{(k)} - \eta\mathbf{G}^{(k)}\right)\mathbf{W}^{(k)}.$$

Multiplying a vector $\mathbf{1}/m$ on both sides, we have

$$\overline{\mathbf{x}}^{(k+1)} = \overline{\mathbf{x}}^{(k)} - \frac{\eta}{m}\mathbf{G}^{(k)}\mathbf{1}.$$

C.2   Lemmas

Lemma 1. *Let $\{\mathbf{W}^{(k)}\}_{k=1}^{\infty}$ be an i.i.d. symmetric and doubly stochastic matrices sequence. The size of each matrix is $m \times m$. Then, for any matrix $\mathbf{B} \in \mathbb{R}^{d\times m}$,*

$$\mathbb{E}\left[\left\|\mathbf{B}\left(\prod_{l=1}^{n}\mathbf{W}^{(l)} - \mathbf{J}\right)\right\|_{F}^{2}\right] \le \rho^n \|\mathbf{B}\|_{F}^{2}$$

where $\rho := \sigma_{max}(\mathbb{E}[\mathbf{W}^{(k)\top}\mathbf{W}^{(k)}] - \mathbf{J}) = \left\|\mathbb{E}[\mathbf{W}^{(k)\top}\mathbf{W}^{(k)}] - \mathbf{J}\right\|_2$.

Proof. For the ease of writing, let us define $\mathbf{A}_{q,n} := \prod_{l=q}^{n}\mathbf{W}^{(l)} - \mathbf{J}$ and use $\mathbf{b}_i^{\top}$ to denote the i-th row vector of $\mathbf{B}$. Since for all $k \in \mathbb{N}$, we have $\mathbf{W}^{(k)\top} = \mathbf{W}^{(k)}$ and $\mathbf{W}^{(k)}\mathbf{J} = \mathbf{J}\mathbf{W}^{(k)} = \mathbf{J}$. Thus, one can obtain

$$\mathbf{A}_{1,n} = \prod_{k=1}^{n}\left(\mathbf{W}^{(k)} - \mathbf{J}\right) = \mathbf{A}_{1,n-1}\left(\mathbf{W}^{(n)} - \mathbf{J}\right).$$

Then, taking the expectation with respect to $\mathbf{W}^{(n)}$,

$$\mathbb{E}_{\mathbf{W}^{(n)}}\left[\|\mathbf{B}\mathbf{A}_{1,n}\|_{F}^{2}\right] = \sum_{i=1}^{d}\mathbb{E}_{\mathbf{W}^{(n)}}\left[\|\mathbf{b}_i^{\top}\mathbf{A}_{1,n}\|^{2}\right]$$

$$= \sum_{i=1}^{d}\mathbb{E}_{\mathbf{W}^{(n)}}\left[\mathbf{b}_i^{\top}\mathbf{A}_{1,n-1}(\mathbf{W}^{(n)\top}\mathbf{W}^{(n)} - \mathbf{J})\mathbf{A}_{1,n-1}^{\top}\mathbf{b}_i\right]$$

$$= \sum_{i=1}^{d}\mathbf{b}_i^{\top}\mathbf{A}_{1,n-1}\mathbb{E}_{\mathbf{W}^{(n)}}\left[(\mathbf{W}^{(n)\top}\mathbf{W}^{(n)} - \mathbf{J})\right]\mathbf{A}_{1,n-1}^{\top}\mathbf{b}_i.$$

Let $\mathbf{C} = \mathbb{E}_{\mathbf{W}^{(n)}}\left[(\mathbf{W}^{(n)\top}\mathbf{W}^{(n)} - \mathbf{J})\right]$ and $\mathbf{v}_i = \mathbf{A}_{1,n-1}^\top \mathbf{b}_i$, then

$$\mathbb{E}_{\mathbf{W}^{(n)}}\left[\|\mathbf{B}\mathbf{A}_{1,n}\|_{\mathrm{F}}^2\right] = \sum_{i=1}^d \mathbf{v}_i^\top \mathbf{C}\mathbf{v}_i$$

$$\leq \sigma_{\max}(\mathbf{C}) \sum_{i=1}^d \mathbf{v}_i^\top \mathbf{v}_i$$

$$= \rho \|\mathbf{B}\mathbf{A}_{1,n-1}\|_{\mathrm{F}}^2.$$

Repeat the following procedure, since $\mathbf{W}^{(k)}$'s are i.i.d. matrices, we have

$$\mathbb{E}_{\mathbf{W}^{(1)}}\ldots\mathbb{E}_{\mathbf{W}^{(n-1)}}\mathbb{E}_{\mathbf{W}^{(n)}}\left[\|\mathbf{B}\mathbf{A}_{1,n}\|_{\mathrm{F}}^2\right] \leq \rho^n \|\mathbf{B}\|_{\mathrm{F}}^2.$$

Here, we complete the proof. $\qquad\square$

C.3   Proof of Theorem 1

Since the objective function $F(\mathbf{x})$ is Liptchitz smooth, it means that

$$F(\overline{\mathbf{x}}^{(k+1)}) - F(\overline{\mathbf{x}}^{(k)}) \leq \left\langle \nabla F(\overline{\mathbf{x}}^{(k)}),\, \overline{\mathbf{x}}^{(k+1)} - \overline{\mathbf{x}}^{(k)} \right\rangle + \frac{L}{2}\left\|\overline{\mathbf{x}}^{(k+1)} - \overline{\mathbf{x}}^{(k)}\right\|^2.$$

Plugging into the update rule $\overline{\mathbf{x}}^{(k+1)} = \overline{\mathbf{x}}^{(k)} - \eta \mathbf{G}^{(k)}\mathbf{1}/m$, we have

$$F(\overline{\mathbf{x}}^{(k+1)}) - F(\overline{\mathbf{x}}^{(k)}) \leq -\eta\left\langle \nabla F(\overline{\mathbf{x}}^{(k)}),\, \frac{\mathbf{G}^{(k)}\mathbf{1}}{m} \right\rangle + \frac{\eta^2 L}{2}\left\|\frac{\mathbf{G}^{(k)}\mathbf{1}}{m}\right\|^2.$$

Then, taking the expectation with respect to random mini-batches at k-th iteration,

$$\mathbb{E}_k\left[F(\overline{\mathbf{x}}^{(k+1)})\right] - F(\overline{\mathbf{x}}^{(k)}) \leq -\eta\left\langle \nabla F(\overline{\mathbf{x}}^{(k)}),\, \frac{\nabla\mathbf{F}^{(k)}\mathbf{1}}{m} \right\rangle + \frac{\eta^2 L}{2}\mathbb{E}_k\left[\left\|\frac{\mathbf{G}^{(k)}\mathbf{1}}{m}\right\|^2\right].$$

For the first term in (19), since $2\langle a,\, b\rangle = \|a\|^2 + \|b\|^2 - \|a - b\|^2$, we have

$$\left\langle \nabla F(\overline{\mathbf{x}}^{(k)}),\, \frac{\nabla\mathbf{F}^{(k)}\mathbf{1}}{m} \right\rangle = \left\langle \nabla F(\overline{\mathbf{x}}^{(k)}),\, \frac{1}{m}\sum_{i=1}^m \nabla F_i(\mathbf{x}_i^{(k)}) \right\rangle \tag{19}$$

$$= \frac{1}{2}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2 + \left\|\frac{1}{m}\sum_{i=1}^m \nabla F_i(\mathbf{x}_i^{(k)})\right\|^2 - \left\|\nabla F(\overline{\mathbf{x}}^{(k)}) - \frac{1}{m}\sum_{i=1}^m \nabla F_i(\mathbf{x}_i^{(k)})\right\|^2\right] \tag{20}$$

Recall that $\nabla F(\overline{\mathbf{x}}^{(k)}) = \frac{1}{m} \sum_{i=1}^{m} \nabla F_i(\overline{\mathbf{x}})$,

$$\left\| \nabla F(\overline{\mathbf{x}}^{(k)}) - \frac{1}{m} \sum_{i=1}^{m} \nabla F_i(\mathbf{x}_i^{(k)}) \right\|^2 = \left\| \frac{1}{m} \sum_{i=1}^{m} \left[\nabla F_i(\overline{\mathbf{x}}^{(k)}) - \nabla F_i(\mathbf{x}_i^{(k)}) \right] \right\|^2 \tag{21}$$

$$\overset{\text{Jensen's Inequality}}{\leq} \frac{1}{m} \sum_{i=1}^{m} \left\| \nabla F_i(\overline{\mathbf{x}}^{(k)}) - \nabla F_i(\mathbf{x}_i^{(k)}) \right\|^2 \tag{22}$$

$$\leq \frac{L^2}{m} \sum_{i=1}^{m} \left\| \overline{\mathbf{x}}^{(k)} - \mathbf{x}_i^{(k)} \right\|^2 \tag{23}$$

where the last inequality follows the Lipschitz smooth assumption. Then, plugging (23) into (20), we obtain

$$\left\langle \nabla F(\overline{\mathbf{x}}^{(k)}), \frac{\nabla \mathbf{F}^{(k)} \mathbf{1}}{m} \right\rangle \geq \frac{1}{2} \left\| \nabla F(\overline{\mathbf{x}}^{(k)}) \right\|^2 + \frac{1}{2} \left\| \frac{\nabla \mathbf{F}^{(k)} \mathbf{1}}{m} \right\|^2 - \frac{L^2}{2m} \left\| \mathbf{X}^{(k)}(\mathbf{I} - \mathbf{J}) \right\|_{\mathrm{F}}^2. \tag{24}$$

Next, for the second part in (19),

$$\mathbb{E}_k \left[\left\| \frac{1}{m} \sum_{i=1}^{m} g_i(\mathbf{x}_i^{(k)}) \right\|^2 \right] = \mathbb{E}_k \left[\frac{1}{m} \sum_{i=1}^{m} \left[g_i(\mathbf{x}_i^{(k)}) - \nabla F_i(\mathbf{x}_i^{(k)}) + \nabla F_i(\mathbf{x}_i^{(k)}) \right] \right]^2 \tag{25}$$

$$= \frac{1}{m^2} \sum_{i=1}^{m} \mathbb{E}_k \left[\left\| g_i(\mathbf{x}_i^{(k)}) - \nabla F_i(\mathbf{x}_i^{(k)}) \right\|^2 \right] + \left\| \frac{1}{m} \sum_{i=1}^{m} \nabla F_i(\mathbf{x}_i^{(k)}) \right\|^2 \tag{26}$$

$$\leq \frac{\sigma^2}{m} + \left\| \frac{\nabla \mathbf{F}^{(k)} \mathbf{1}}{m} \right\|^2 \tag{27}$$

where the last inequality is according to the bounded variance assumption. Then, combining (24) and (27) and taking the total expectation over all random variables, one can obtain:

$$\mathbb{E} \left[F(\overline{\mathbf{x}}^{(k+1)}) - F(\overline{\mathbf{x}}^{(k)}) \right] \leq -\frac{\eta}{2} \mathbb{E} \left[\left\| \nabla F(\overline{\mathbf{x}}^{(k)}) \right\|^2 \right] - \frac{\eta}{2}(1 - \eta L) \mathbb{E} \left[\left\| \frac{\nabla \mathbf{F}^{(k)} \mathbf{1}}{m} \right\|^2 \right] +$$

$$\frac{\eta L^2}{2m} \mathbb{E} \left[\left\| \mathbf{X}^{(k)}(\mathbf{I} - \mathbf{J}) \right\|_{\mathrm{F}}^2 \right] + \frac{\eta^2 L \sigma^2}{2m}.$$

Summing over all iterates and taking the average,

$$\frac{\mathbb{E} \left[F(\overline{\mathbf{x}}^{K}) - F(\overline{\mathbf{x}}^{(1)}) \right]}{K} \leq -\frac{\eta}{2} \frac{1}{K} \sum_{i=1}^{K} \mathbb{E} \left[\left\| \nabla F(\overline{\mathbf{x}}^{(k)}) \right\|^2 \right] - \frac{\eta}{2}(1 - \eta L) \frac{1}{K} \sum_{k=1}^{K} \mathbb{E} \left[\left\| \frac{\nabla \mathbf{F}^{(k)} \mathbf{1}}{m} \right\|^2 \right] +$$

$$\frac{\eta L^2}{2mK} \sum_{i=1}^{K} \mathbb{E} \left[\left\| \mathbf{X}^{(k)}(\mathbf{I} - \mathbf{J}) \right\|_{\mathrm{F}}^2 \right] + \frac{\eta^2 L \sigma^2}{2m}.$$

By minor rearranging, we get

$$
\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^{2}\right] \leq \frac{2\mathbb{E}\left[F(\overline{\mathbf{x}}^{(1)}) - F(\overline{\mathbf{x}}^{(K)})\right]}{\eta K} - \frac{1-\eta L}{m}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\frac{\nabla \mathbf{F}^{(k)}\mathbf{1}}{m}\right\|^{2}\right] +
$$

$$
\frac{L^2}{mK}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^{2}\right] + \frac{\eta L\sigma^2}{m} \tag{28}
$$

$$
\leq \frac{2\left[F(\overline{\mathbf{x}}^{(1)}) - F_{\mathrm{inf}}\right]}{\eta K} - \frac{1-\eta L}{m}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\frac{\nabla \mathbf{F}^{(k)}\mathbf{1}}{m}\right\|^{2}\right] +
$$

$$
\frac{L^2}{mK}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^{2}\right] + \frac{\eta L\sigma^2}{m}. \tag{29}
$$

Now we complete the first part of the proof. Then, we're going to show that the discrepancies among local models $\mathbb{E}\left[\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^{2}\right]$ is upper bounded. According to the update rule of decentralized SGD and the special property of gossip matrix $\mathbf{W}^{(k)}\mathbf{J} = \mathbf{J}\mathbf{W}^{(k)} = \mathbf{J}$, we have

$$
\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J}) = \left(\mathbf{X}^{(k-1)} - \eta\mathbf{G}^{(k-1)}\right)\mathbf{W}^{(k-1)}(\mathbf{I}-\mathbf{J})
$$

$$
= \mathbf{X}^{(k-1)}(\mathbf{I}-\mathbf{J})\mathbf{W}^{(k-1)} - \eta\mathbf{G}^{(k-1)}\mathbf{W}^{(k-1)}(\mathbf{I}-\mathbf{J})
$$

$$
\vdots
$$

$$
= \mathbf{X}^{(1)}(\mathbf{I}-\mathbf{J})\prod_{q=1}^{k-1}\mathbf{W}^{(q)} - \eta\sum_{q=1}^{k-1}\mathbf{G}^{(q)}\left(\prod_{l=q}^{k-1}\mathbf{W}^{(l)} - \mathbf{J}\right).
$$

Since all local models are initiated at the same point, $\mathbf{X}^{(1)}(\mathbf{I}-\mathbf{J}) = 0$. Thus, we can obtain

$$
\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^{2} = \eta^2 \left\|\sum_{q=1}^{k-1}\mathbf{G}^{(q)}\left(\prod_{l=q}^{k-1}\mathbf{W}^{(l)} - \mathbf{J}\right)\right\|_{\mathrm{F}}^{2} \tag{30}
$$

$$
= \eta^2 \left\|\sum_{q=1}^{k-1}\left(\mathbf{G}^{(q)} - \nabla\mathbf{F}^{(q)} + \nabla\mathbf{F}^{(q)}\right)\left(\prod_{l=q}^{k-1}\mathbf{W}^{(l)} - \mathbf{J}\right)\right\|_{\mathrm{F}}^{2} \tag{31}
$$

$$
\leq 2\eta^2 \underbrace{\left\|\sum_{q=1}^{k-1}\left(\mathbf{G}^{(q)} - \nabla\mathbf{F}^{(q)}\right)\left(\prod_{l=q}^{k-1}\mathbf{W}^{(l)} - \mathbf{J}\right)\right\|_{\mathrm{F}}^{2}}_{T_1} + 2\eta^2 \underbrace{\left\|\sum_{q=1}^{k-1}\nabla\mathbf{F}^{(q)}\left(\prod_{l=q}^{k-1}\mathbf{W}^{(l)} - \mathbf{J}\right)\right\|_{\mathrm{F}}^{2}}_{T_2}.
$$

$$
\tag{32}
$$

For the first term T_1 in (32), we have

$$\mathbb{E}\left[T_1\right] = \sum_{q=1}^{k-1} \mathbb{E}\left[\left\|\left(\mathbf{G}^{(q)} - \nabla\mathbf{F}^{(q)}\right)\left(\prod_{l=q}^{k-1}\mathbf{W}^{(l)} - \mathbf{J}\right)\right\|_{\mathrm{F}}^2\right] \tag{33}$$

$$\leq \sum_{q=1}^{k-1} \rho^{k-q}\mathbb{E}\left[\left\|\mathbf{G}^{(q)} - \nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] \tag{34}$$

$$\leq m\sigma^2\rho\left(1 + \rho + \rho^2 + \cdots + \rho^{k-2}\right) \tag{35}$$

$$\leq \frac{m\sigma^2\rho}{1-\rho} \tag{36}$$

where (34) comes from Lemma 1. For the second term T_2 in (32), define $\mathbf{A}_{q,p} = \prod_{l=q}^{p}\mathbf{W}^{(l)} - \mathbf{J}$. Then,

$$\mathbb{E}\left[T_2\right] = \sum_{q=1}^{k-1}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\mathbf{A}_{q,k-1}\right\|_{\mathrm{F}}^2\right] + \sum_{q=1}^{k-1}\sum_{p=1,p\neq q}^{k-1}\mathbb{E}\left[\mathrm{Tr}\{\mathbf{A}_{q,k-1}^{\top}\nabla\mathbf{F}^{(q)\top}\nabla\mathbf{F}^{(p)}\mathbf{A}_{p,k-1}\}\right] \tag{37}$$

$$\leq \sum_{q=1}^{k-1}\rho^{k-q}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] + \sum_{q=1}^{k-1}\sum_{p=1,p\neq q}^{k-1}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\mathbf{A}_{q,k-1}\right\|_{\mathrm{F}}\left\|\nabla\mathbf{F}^{(p)}\mathbf{A}_{p,k-1}\right\|_{\mathrm{F}}\right] \tag{38}$$

$$\leq \sum_{q=1}^{k-1}\rho^{k-q}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] + \sum_{q=1}^{k-1}\sum_{p=1,p\neq q}^{k-1}\mathbb{E}\left[\frac{1}{2\epsilon}\left\|\nabla\mathbf{F}^{(q)}\mathbf{A}_{q,k-1}\right\|_{\mathrm{F}}^2 + \frac{\epsilon}{2}\left\|\nabla\mathbf{F}^{(p)}\mathbf{A}_{p,k-1}\right\|_{\mathrm{F}}^2\right] \tag{39}$$

$$\leq \sum_{q=1}^{k-1}\rho^{k-q}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] + \sum_{q=1}^{k-1}\sum_{p=1,p\neq q}^{k-1}\mathbb{E}\left[\frac{\rho^{k-q}}{2\epsilon}\left\|\nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2 + \frac{\rho^{k-p}\epsilon}{2}\left\|\nabla\mathbf{F}^{(p)}\right\|_{\mathrm{F}}^2\right] \tag{40}$$

where (39) follows Young's Inequality: $2ab \leq a^2/\epsilon + \epsilon b^2, \forall\epsilon > 0$ and (40) follows Lemma 1. Set $\epsilon = \rho^{\frac{p-q}{2}}$, then we have

$$\mathbb{E}\left[T_2\right] \leq \sum_{q=1}^{k-1}\rho^{k-q}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] + \frac{1}{2}\sum_{q=1}^{k-1}\sum_{p=1,p\neq q}^{k-1}\sqrt{\rho}^{2k-p-q}\cdot\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2 + \left\|\nabla\mathbf{F}^{(p)}\right\|_{\mathrm{F}}^2\right] \tag{41}$$

$$= \sum_{q=1}^{k-1}\rho^{k-q}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] + \sum_{q=1}^{k-1}\left[\sqrt{\rho}^{k-q}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right]\cdot\sum_{p=1,p\neq q}^{k-1}\sqrt{\rho}^{k-p}\right] \tag{42}$$

$$= \sum_{q=1}^{k-1}\rho^{k-q}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] + \sum_{q=1}^{k-1}\left[\sqrt{\rho}^{k-q}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right]\cdot\left(\sum_{p=1}^{k-1}\sqrt{\rho}^{k-p} - \sqrt{\rho}^{k-q}\right)\right] \tag{43}$$

$$\leq \frac{\sqrt{\rho}}{1-\sqrt{\rho}}\sum_{q=1}^{k-1}\sqrt{\rho}^{k-q}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right]. \tag{44}$$

Combining (36) and (44) together,

$$\frac{1}{mK}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^{2}\right]\leq\frac{2\eta^{2}\sigma^{2}\rho}{1-\rho}+\frac{2\eta^{2}}{m}\frac{\sqrt{\rho}}{1-\sqrt{\rho}}\frac{1}{K}\sum_{k=1}^{K}\sum_{q=1}^{k-1}\sqrt{\rho}^{k-q}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^{2}\right] \quad (45)$$

$$=\frac{2\eta^{2}\sigma^{2}\rho}{1-\rho}+\frac{2\eta^{2}}{m}\frac{\sqrt{\rho}}{1-\sqrt{\rho}}\frac{1}{K}\sum_{k=1}^{K}\left[\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(k)}\right\|_{\mathrm{F}}^{2}\right]\sum_{q=1}^{K-k}\sqrt{\rho}^{q}\right] \quad (46)$$

$$\leq\frac{2\eta^{2}\sigma^{2}\rho}{1-\rho}+\frac{2\eta^{2}}{m}\frac{\sqrt{\rho}}{1-\sqrt{\rho}}\frac{1}{K}\sum_{k=1}^{K}\left[\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(k)}\right\|_{\mathrm{F}}^{2}\right]\frac{\sqrt{\rho}}{1-\sqrt{\rho}}\right] \quad (47)$$

$$=\frac{2\eta^{2}\sigma^{2}\rho}{1-\rho}+\frac{2\eta^{2}}{m}\frac{\rho}{(1-\sqrt{\rho})^{2}}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(k)}\right\|_{\mathrm{F}}^{2}\right] \quad (48)$$

Note that

$$\left\|\nabla\mathbf{F}^{(k)}\right\|_{\mathrm{F}}^{2}=\sum_{i=1}^{m}\left\|\nabla F_{i}(\mathbf{x}_{i}^{(k)})\right\|^{2} \quad (49)$$

$$=\sum_{i=1}^{m}\left\|\nabla F_{i}(\mathbf{x}_{i}^{(k)})-\nabla F(\mathbf{x}_{i}^{(k)})+\nabla F(\mathbf{x}_{i}^{(k)})-\nabla F(\overline{\mathbf{x}}^{(k)})+\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^{2} \quad (50)$$

$$\leq 3\sum_{i=1}^{m}\left[\left\|\nabla F_{i}(\mathbf{x}_{i}^{(k)})-\nabla F(\mathbf{x}_{i}^{(k)})\right\|^{2}+\left\|\nabla F(\mathbf{x}_{i}^{(k)})-\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^{2}+\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^{2}\right] \quad (51)$$

$$\leq 3m\zeta^{2}+3L^{2}\sum_{i=1}^{m}\left\|\mathbf{x}_{i}^{(k)}-\overline{\mathbf{x}}^{(k)}\right\|^{2}+3m\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^{2} \quad (52)$$

$$=3m\zeta^{2}+3L^{2}\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^{2}+3m\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^{2}. \quad (53)$$

Plugging (53) back into (48), we have

$$\frac{1}{mK}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^{2}\right]\leq\frac{2\eta^{2}\sigma^{2}\rho}{1-\rho}+\frac{6\eta^{2}\zeta^{2}\rho}{(1-\sqrt{\rho})^{2}}+\frac{6\eta^{2}L^{2}\rho}{(1-\sqrt{\rho})^{2}}\frac{1}{mK}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^{2}\right]+$$

$$\frac{6\eta^{2}\rho}{(1-\sqrt{\rho})^{2}}\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^{2}\right].$$

After minor rearranging, we get

$$\frac{1}{mK}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^{2}\right]\leq\frac{1}{1-D}\left[\frac{2\eta^{2}\sigma^{2}\rho}{1-\rho}+\frac{6\eta^{2}\zeta^{2}\rho}{(1-\sqrt{\rho})^{2}}+\frac{6\eta^{2}\rho}{(1-\sqrt{\rho})^{2}}\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^{2}\right]\right]$$

$$(54)$$

where $D = \frac{6\eta^2 L^2 \rho}{(1-\sqrt{\rho})^2}$. Then, plugging (54) back into (29), we have

$$\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right] \leq \frac{2[F(\overline{\mathbf{x}}^{(1)}) - F_{\text{inf}}]}{\eta K} + \frac{\eta L \sigma^2}{m} + \frac{1}{1-D}\frac{2\eta^2 L^2 \sigma^2 \rho}{1-\rho} + \frac{D\zeta^2}{1-D} + \frac{D}{1-D}\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right].$$

It follows that

$$\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right] \leq \left(\frac{2[F(\overline{\mathbf{x}}^{(1)}) - F_{\text{inf}}]}{\eta K} + \frac{\eta L \sigma^2}{m}\right)\frac{1-D}{1-2D} + \left(\frac{2\eta^2 L^2 \sigma^2 \rho}{1-\rho} + \frac{6\eta^2 L^2 \zeta^2 \rho}{(1-\sqrt{\rho})^2}\right)\frac{1}{1-2D} \tag{55}$$

$$= \left(\frac{2[F(\overline{\mathbf{x}}^{(1)}) - F_{\text{inf}}]}{\eta K} + \frac{\eta L \sigma^2}{m}\right)\frac{1-D}{1-2D} + \frac{2\eta^2 L^2 \rho}{1-\sqrt{\rho}}\left(\frac{\sigma^2}{1+\sqrt{\rho}} + \frac{3\zeta^2}{1-\sqrt{\rho}}\right)\frac{1}{1-2D} \tag{56}$$

$$\leq \left(\frac{2[F(\overline{\mathbf{x}}^{(1)}) - F_{\text{inf}}]}{\eta K} + \frac{\eta L \sigma^2}{m}\right)\frac{1}{1-2D} + \frac{2\eta^2 L^2 \rho}{1-\sqrt{\rho}}\left(\frac{\sigma^2}{1+\sqrt{\rho}} + \frac{3\zeta^2}{1-\sqrt{\rho}}\right)\frac{1}{1-2D} \tag{57}$$

Recall that we require that $\eta L \leq (1-\sqrt{\rho})/4\sqrt{\rho}$. Therefore,

$$D = \frac{6\eta^2 L^2 \rho}{(1-\sqrt{\rho})^2} \leq \frac{3}{8} < \frac{1}{2} \Rightarrow \frac{1}{1-2D} \leq 4.$$

Plugging the upper bound of D into (57) and set $\eta = \sqrt{\frac{m}{K}}$, we have

$$\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right] \leq \frac{8[F(\overline{\mathbf{x}}^{(1)}) - F_{\text{inf}}] + 4L\sigma^2}{\sqrt{mK}} + \frac{8m}{K}\frac{L^2 \rho}{1-\sqrt{\rho}}\left(\frac{\sigma^2}{1+\sqrt{\rho}} + \frac{3\zeta^2}{1-\sqrt{\rho}}\right)$$

$$= \mathcal{O}\left(\frac{1}{\sqrt{mK}}\right) + \mathcal{O}\left(\frac{m}{K}\right).$$

Here we complete the proof.

D   Spectral Graph Theory

The inter-agent communication network is a simple[2] undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where V denotes the set of agents or vertices with cardinality $|\mathcal{V}| = m$, and $\mathcal{E}$ the set of edges. If there exists an edge between agents i and j, then $(i,j) \in E$. A path between agents i and j of length n is a sequence $(i = p_0, p_1, \cdots, p_n = j)$ of vertices, such that $(p_t, p_{t+1}) \in \mathcal{E}$, $0 \leq t \leq n-1$. A graph is connected if there exists a path between all possible agent pairs. The neighborhood of an agent n is given by $\Omega_n = \{j \in \mathcal{V}|(n,j) \in \mathcal{E}\}$. The degree of agent n is given by $d_n = |\Omega_n|$. The structure of the graph is

---

[2]A graph is said to be simple if it is devoid of self loops and multiple edges.

represented by the symmetric $m \times m$ adjacency matrix $\mathbf{A} = [A_{ij}]$, where $A_{ij} = 1$ if $(i, j) \in E$, and 0 otherwise. The degree matrix is given by the diagonal matrix $\mathbf{D} = diag(d_1 \cdots d_m)$. The graph Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. The Laplacian is a positive semidefinite matrix, hence its eigenvalues can be ordered and represented as $0 = \lambda_1(\mathbf{L}) \leq \lambda_2(\mathbf{L}) \leq \cdots \lambda_m(\mathbf{L})$. Furthermore, a graph is connected if and only if $\lambda_2(\mathbf{L}) > 0$