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

May 22, 2019

Abstract

The trade-off between convergence error and communication delays in decentralized stochastic gradient descent (SGD) is dictated by the sparsity of the inter-worker communication graph. In this paper, we propose Matcha, a decentralized SGD method where we use matching decomposition sampling of the base graph to parallelize inter-worker information exchange so as to significantly reduce communication delay. At the same time, under standard assumptions for any general topology, in spite of the significant reduction of the communication delay, Matcha maintains the same convergence rate as that of the state-of-the-art in terms of epochs. Experiments on a suite of datasets and deep neural networks validate the theoretical analysis and demonstrate the effectiveness of the proposed scheme as far as reducing communication delays is concerned.

1 Introduction

Due to the enormity of the training data used today, distributing the data and the computation over a network of worker nodes has attracted intensive research efforts in recent years. In this paper, we will focus on parallelizing synchronous SGD in a decentralized setting without central coordinators (i.e., parameter server). Given an arbitrary network topology, all nodes can only exchange parameters or gradients with their local neighbors. This scenario is common and useful when training in large-scale sensor networks, multi-agent systems, as well as federated learning in edge devices.

Error-Runtime Trade-off in Decentralized SGD. In the context of decentralized optimization, previous works have studied the error convergence in terms of iterations or communication rounds for decentralized gradient descent [22, 7, 40, 42, 10, 27, 30, 13] mostly for (strongly) convex loss functions. Recent works have extended the analysis to decentralized SGD for non-convex loss functions and subsequently applied it to distributed deep learning in both synchronous [17, 11, 35] and asynchronous settings [1, 18]. However, most of existing works do not explicitly consider how the topology affects the runtime, that is, *wall-clock time required to complete each SGD iteration.* Well-connected networks encourage faster consensus and give better mean square error convergence rates, but they incur communication delays which increases with increasing node degree. To strike the best error-runtime trade-off, one can carefully design the network topology, for example, using expander graphs that are sparse while being well connected [6, 23]. However, systems constraints such as locality may preclude us from designing such arbitrary network topologies. Other approaches for optimizing the per-epoch rate of convergence of decentralized procedures through efficient link-scheduling or constraining the number of allowable links, have been proposed [4]. However, these design criteria do not take into account the wall-clock time which depend on the parallel versus sequential scheduling of the communication links which we describe later. This raises a pertinent question: for a given topology of worker nodes, how can we achieve the fastest convergence in terms of mean square error versus wall-clock time for a synchronous decentralized SGD algorithm?

Related Works. There have been massive amount of work in the context of algorithms [37, 33, 36, 5, 26] and systems [16, 43, 9] that improve the communication efficiency of synchronous distributed SGD in a fully-connected network. However, it is still unclear whether theses strategies can be directly applied to any general decentralized setting. Given an arbitrary network topology, recent works [14, 29] propose to compress the transmitted message size to reduce the communication bandwidth. However, these methods may not help if the network latency (i.e., time to establish handshakes) is high. Other communication efficient schemes [25, 31], which focus on reducing the number of communications by sparsifying communications over time have also been proposed which do not take into account communication delays. Instead, we focus on a complementary idea of reducing the effective node degree so as to reduce the communication delay, which is suitable for both high latency and low bandwidth settings and can be easily combined with existing compression schemes.

Our Proposed Method Matcha. In this paper, we propose Matcha, a decentralized SGD method based on matching decomposition sampling, that drastically *reduces the communication delay per iteration for any given node topology while maintaining the same error convergence speed.* The following key ideas allow us to achieve this: 1) we decompose the graph topology into matchings consisting of disjoint communication links that can operate in parallel and save communication delay, and 2) in each iteration, we carefully sample a subset of these matchings to construct a sparse subgraph of the base topology, 3) this sequence of subgraphs results in more frequent communication over connectivity-critical links (ensuring fast error convergence) and less frequent communication over other links (saving communication delays).

An illustration of the advantages of using Matcha is presented in Figure 1. It shows that the reduction of communication complexity at different nodes is not uniform. In particular, when the communication budget is 0.5 (using 50% time to communicate at each iteration compared to vanilla decentralized SGD), critical links (such as edge $(0, 4)$) end up being used for communication with high priority. As a result, the communication time at a degree 1 node (node 4 for example) does not change. On the other hand, links, which are incident to the busiest node, will be used for communication infrequently. The communication time at a node of degree 5 (node 1 for example) is directly reduced by half, as it is the bottleneck of run time per iteration. We further validate the effectiveness of Matcha through theoretical analysis and extensive experiments (see Sections 4 and 5). Besides a win-win in the wall clock time-error trade-off, Matcha has many more practical benefits. First, the proposed algorithm is simple, in the sense that the communication schedule (i.e., the sequence of sparse subgraphs) of Matcha can be obtained apriori. There is no additional runtime overhead during training. Furthermore, Matcha provides a highly flexible communication scheme among nodes. By setting the communication budget, one can easily tailor the communication time to various system and problem settings, allowing a better trade-off between communication and computation. In our experiments on CIFAR-100, Matcha gets a 50x reduction in communication

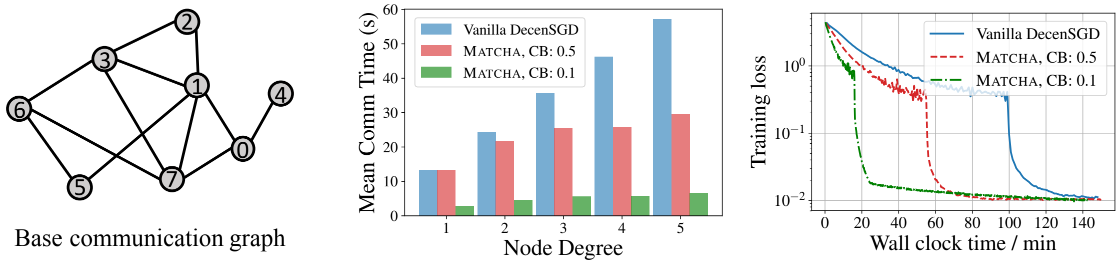


Figure 1: Comparison between vanilla decentralized SGD (DecenSGD) and Matcha when training WideResNet [41] on CIFAR-100 [15]. In vanilla DecenSGD, all links are used for communication at every iteration. CB stands for communication budget: the ratio of expected communication time between Matcha and vanilla DecenSGD.

delay per iteration, and up to 5x reduction in wall-clock time to achieve the same training accuracy.

2 Problem Formulation and Preliminaries

Consider a network of m worker nodes. The communication links connecting the workers 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 with its neighbors, that is, it can communicate with node j only if $(i, j) \in \mathcal{E}$.
Furthermore, 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:

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

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 (or consensus-based distributed SGD) [32, 22, 40, 42, 11, 17, 10] is an effective way to optimize the empirical risk (1) in the considered setting. The algorithm alternates between the consensus and gradient steps 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 uniformly at random 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 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., $(i,j) \in \mathcal{E}$. In order

[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.

to guarantee that all nodes can reach consensus and converge to a common stationary point, mixing matrix $\mathbf{W}$ can be taken to be symmetric and doubly stochastic. For instance, if node 1 only connects with node 2 and 3, then the first row of $\mathbf{W}$ can be $[1-2\alpha, \alpha, \alpha, 0, \ldots, 0]$, where α is constant.

Convergence in terms of Error Versus Wallclock Time. The total training time of an optimization algorithm is a product of two factors: 1) total iterations; and 2) run time per iteration. In a decentralized setup involving multiple worker nodes without a coordinating master node, both of these two factors are closely related to the graph topology. While there has been extensive literature studying the first factor [22, 7, 21], the second factor is less explored from a theoretical point of view.

In DecenSGD, each node needs to communicate with all of its neighbors at each iteration. The node with the highest degree in the graph (the busiest node) turns out to be the bottleneck as far as reducing communication time to finish one consensus step is concerned. Intuitively, the communication time per iteration monotonically increases with the maximal node degree. In general, the scaling is linear as commonly used in previous works [9, 31, 6, 28, 18], since the bandwidth is limited and both the total transmitted message size and number of handshakes are linear in the degree of the node. In this paper, we will focus on this linear scaling delay model, but the main idea can also be extended to other scaling rules. Without loss of generality, we assume the communication (sending and receiving model parameters) over one link costs 1 unit of time. Then, the communication per iteration takes at least the maximal degree $\Delta(\mathcal{G})$ units of time. Although a denser base graph may require less iterations to converge, it consumes more communication time, resulting in longer training time.

Preliminaries on Graph Theory. The communication graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ can be abstracted by 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 $\mathbf{L} = \operatorname{diag}(d_1, \ldots, d_m) - \mathbf{A}$, 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* [2]. A larger value of λ_2 implies a denser graph. Moreover, we will use the notion of matching, defined as follows.

Definition 1 (Matching)**.** A matching in $\mathcal{G}$ is a subgraph of $\mathcal{G}$, in which each vertex is incident with at most one edge.

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. A brief illustration is also shown in Figure 2.

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 $\mathcal{E}_i \bigcap \mathcal{E}_j = \phi, \forall i \neq j$. This decomposition procedure can be achieved via Misra & Gries edge coloring algorithm [20], which guarantees that the number of disjoint matchings M equals to either $\Delta(\mathcal{G})$ or $\Delta(\mathcal{G}) + 1$, where $\Delta(\mathcal{G})$ is the maximal degree of graph $\mathcal{G}$.

The main benefit of using matchings is that it allows parallel communication, due to the disjoint links connecting nodes. Recall that a matching is a set of edges without common vertices. In each matching, nodes have at most one neighbor. Thus, all edges (or links) can be used to communicate over in parallel. The communication time for each matching is exactly 1 unit. Inspired by this

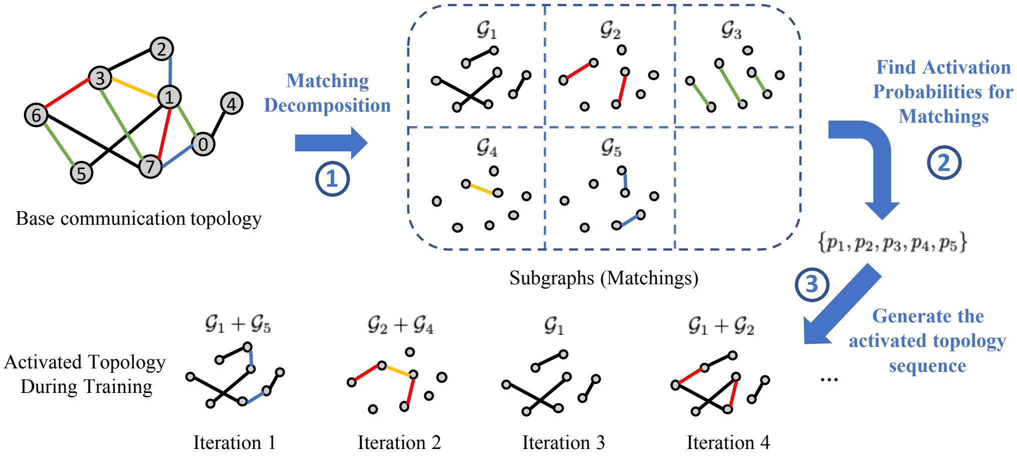

2
3
1
4
6
0
5
7
Base communication topology
Matching Decomposition
1
$\mathcal{G}_1$
$\mathcal{G}_2$
$\mathcal{G}_3$
$\mathcal{G}_4$
$\mathcal{G}_5$
Subgraphs (Matchings)
Find Activation Probabilities for Matchings
2
$\{p_1, p_2, p_3, p_4, p_5\}$
3
Generate the activated topology sequence
Activated Topology During Training
$\mathcal{G}_1 + \mathcal{G}_5$
$\mathcal{G}_2 + \mathcal{G}_4$
$\mathcal{G}_1$
$\mathcal{G}_1 + \mathcal{G}_2$
...
Iteration 1
Iteration 2
Iteration 3
Iteration 4

Figure 2: Illustration of the proposed method. Given the base communication graph, we decompose it into disjoint subgraphs (in particular, matchings). Then, at each communication round, subgraph $\mathcal{G}_j$ is activated with probability p_j. Worker nodes are synchronized only through the activated topology. On the contrary, vanilla DecenSGD uses the base communication topology at every iterations.

matching decomposition scheme, communicating over all matchings sequentially is a simple and efficient way to implement the consensus step in decentralized training algorithm. The total communication time will be linear in the number of matchings and be bounded by $(\Delta(\mathcal{G}) + 1)$ units, which matches with the communication time model discussed in Section 2 and previous works [31, 6, 28, 18].

Step 2: Computing Matching Activation Probabilities. In order to control the communication time, we assign each matching a Bernoulli random variable B_j, which is 1 with probability p_j and 0 otherwise, $\forall j \in \{1, \dots, M\}$. Then, at each iteration, only when the realization of B_j is 1, links in the corresponding matching will be used for information exchange between the corresponding worker nodes. As a result, when B_j's are independent to each other, the communication time per iteration can be written as

$$\text{Expected Communication Time} = \mathbb{E}\left[\sum_{j=1}^{M} \mathsf{B}_j\right] = \sum_{j=1}^{M} p_j. \tag{3}$$

We define p_j as the *activation probability*. By controlling the summation of all activation probabilities, one can easily change the expected communication time. When all p_j's equal to 1, the algorithm reduces to that of vanilla DecenSGD and takes M units of time to finish one consensus step. We further define *communication budget (CB)*, in terms of fraction of communication time of vanilla DecenSGD (e.g., CB = 0.2 means using only 20% communication time per iteration of vanilla DecenSGD). Given a CB, there can be many feasible activation probabilities. As mentioned before, a key contribution of this paper is that we give more importance to critical links. This is achieved by controlling the activation probabilities for the matchings. Formally, we choose a set of activation probabilities by solving the following optimization problem:

$$\begin{aligned} \max_{p_1,\ldots,p_M} \quad & \lambda_2\left(\textstyle\sum_{j=1}^{M} p_j \mathbf{L}_j\right) \\ \text{subject to} \quad & \textstyle\sum_{j=1}^{M} p_j \leq \mathrm{CB} \cdot M,\ 0 \leq p_j \leq 1,\ \forall j \in \{1,2,\ldots,M\}, \end{aligned} \tag{4}$$

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. CB is the pre-determined communication budget. Moreover, recall that λ_2 represents the algebraic connectivity and is a concave function [12, 2]. Thus, it directly follows that (4) is a convex problem and can be solved efficiently.

Step 3: Generating Random Topology Sequence. At the k-th iteration, the communication among nodes only happen over links in the activated topology $\mathcal{G}^{(k)} = \bigcup_{j=1}^{M} \mathsf{B}_j^{(k)} \mathcal{G}_j$, which is sparse or even disconnected. According to this activated topology, we need to further specify in what proportions the local models are averaged together in order to perform the consensus step in (2). A common practice is to use an equal weight matrix [38, 12, 7] as follows:

$$\mathbf{W}^{(k)} = \mathbf{I} - \alpha \mathbf{L}^{(k)} = \mathbf{I} - \alpha \sum_{j=1}^{M} \mathsf{B}_j^{(k)} \mathbf{L}_j, \tag{5}$$

where $\mathbf{L}^{(k)}$ denotes the graph Laplacian at the k-th iteration. The matrix $\mathbf{W}^{(k)}$ is symmetric and doubly stochastic by construction. The parameter α represents the weight of neighbor's information in the consensus step. By setting a proper value of α, the convergence of MATCHA to a stationary point can be guaranteed. In particular, we select a value of α that minimizes the optimization error upper bound. In Section 4 Lemma 1, we will show that optimizing α can be formulated as a semi-definite programming problem. It needs to be solved only once at the beginning of training.

Extension to Other Design Choices. To sum up, the inputs of the proposed algorithm MATCHA are: a base communication topology $\mathcal{G}$ and a target communication budget CB. Then, following the steps 1 to 3, the algorithm will output a random topology sequence $\{\mathcal{G}^{(k)}\}_{k=1}^{\infty}$ and a value of α that defines the inter-node information exchange. All of these information can be obtained and assigned apriori to worker nodes before starting the training procedure.

We note that the framework involving randomly activating subgraphs, is very general and can be extended to various other delay models and graph decomposition methods. For example, 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 (3) accordingly. Moreover, rather than matching decomposition, it is also possible to decompose the base topology into subgraphs of different types. For instance, each subraph can be a single edge in the base graph $\mathcal{G}$.

Among all possible variants, we would like to highlight one special case: *Periodic DecenSGD (P-DecenSGD)*, which has appeared in previous works [31, 35]. In P-DecenSGD, all links in the base topology are activated together ($\mathsf{B}_1 = \cdots = \mathsf{B}_M = 1$) after every few iterations. In this case, the communication budget is equivalent to communication frequency. In Sections 4 and 5, we will use P-DecenSGD as another benchmark for comparison.

4 Theoretical Analyses

In this section, we provide convergence guarantees for MATCHA. To be specific, we first provide convergence guarantees where we explicitly quantify the dependence of the mean square error on

the arbitrary random topology sequence. Then, in Section 4.2, we analyze the spectral norm of the random topology generated by MATCHA. All proofs are provided in the Appendix.
In order to facilitate the analysis, we define the averaged iterate as $\overline{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i$ 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 convergence analysis is centered around the following assumptions, which are common in distributed optimization literature [3, 22, 17].

Assumption 1. *Each local objective function $F_i(\mathbf{x})$ 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 is an unbiased estimator of the true gradient of 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 sources of randomness upto time k, i.e., sigma algebra generated by noise of the stochastic gradients and the graph activation probabilities before iteration k.*

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

4.1 Convergence Analysis for Arbitrary Random Topology

Theorem 1 (**Basic Convergence Result**). *Suppose that all local models are initiated at the same iterate $\overline{\mathbf{x}}^{(1)}$ and $\{\mathbf{W}^{(k)}\}_{k=1}^{K}$ is an i.i.d. random matrix sequence. Then, under Assumptions 1 to 3, if the learning rate satisfies $\eta L \leq 1$, then after total K iterations, we have that,*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right] \leq \underbrace{\frac{2[F(\overline{\mathbf{x}}^{(1)}) - F_{inf}]}{\eta K} + \frac{\eta L \sigma^2}{m}}_{centralized\ SGD} + \eta^2 L^2 \sigma^2 \frac{2\rho}{1-\rho}$$
$$+ \eta^2 L^2 \frac{2\rho}{(1-\sqrt{\rho})^2}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m}\left\|\nabla F_i(\mathbf{x}_i^{(k)})\right\|^2\right], \tag{6}$$

where ρ is the spectral norm (i.e., largest singular value) of matrix $\mathbb{E}[\mathbf{W}^{(k)\top}\mathbf{W}^{(k)}] - \frac{1}{m}\mathbf{1}\mathbf{1}^\top$.

The result in Theorem 1 can be further refined by introducing new assumptions on the dissimilarities among local objectives. For the brevity of result, we simply assume the local gradients are uniformly bounded ($\|\nabla F_i(\mathbf{x})\|^2 \leq D$) as [7, 39, 14] and derive the following corollary. In the Appendix, we provide another version of corollary with weaker assumption as in [17].

Corollary 1. *Suppose for each local objective, we have $\|\nabla F_i(\mathbf{x})\|^2 \leq D$ and the learning rate is set as $\eta = \frac{1}{L}\sqrt{\frac{m}{K}}$, then after total K iterations,*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right] \leq \frac{2L[F(\overline{\mathbf{x}}^{(1)}) - F_{inf}] + \sigma^2}{\sqrt{mK}} + \frac{2m\rho}{K}\left[\frac{\sigma^2}{1-\rho} + \frac{D}{(1-\sqrt{\rho})^2}\right]$$
$$= \mathcal{O}\left(\frac{1}{\sqrt{mK}} + \frac{m}{K}\frac{\rho}{(1-\sqrt{\rho})^2}\right), \tag{7}$$

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

Dependence on the Random Topology. Theorem 1 together with Corollary 1 show that when the other algorithm parameters are fixed, the mean square error monotonically increases with ρ. Typically, the value of spectral norm relates to the connectivity of the random topology. If the activated topology is fully connected, i.e., $\mathbf{W}^{(k)} = \mathbf{1}\mathbf{1}^\top/m$, then $\rho = 0$ and Theorem 1 recovers the convergence results for centralized SGD. However, if there are two groups of nodes which are not connected during the whole training procedure, then $\rho = 1$. Local models cannot achieve consensus and the iterates will diverge. Since in Matcha, we optimize the connectivity of the average activated topology, it is important to guarantee $\rho < 1$. We further prove this statement in Theorem 2.

4.2 Analysis for Random Topology Sequence Generated by Matcha

Theorem 2 (**Existence Proof**)**.** *Suppose the base graph $\mathcal{G}$ is connected. Let $\mathbf{L}^{(k)}$ denote the Laplacian matrix of the activated topology at k-th iteration in* Matcha*. If the mixing matrix is defined as $\mathbf{W}^{(k)} = \mathbf{I} - \alpha\mathbf{L}^{(k)}$, then there exists a value of α such that $\rho = \left\|\mathbb{E}[\mathbf{W}^{(k)\top}\mathbf{W}^{(k)} - \frac{1}{m}\mathbf{1}\mathbf{1}^\top]\right\|_2 < 1$.*

Theorem 2 and Corollary 1 guarantee the convergence of Matcha. When the communication budget (or activation probability) varies, the value of α should be changed. However, finding an optimal value of α, which minimizes the spectral norm, is not trivial. It is hard to get the analytic form of α. However, we show that optimizing α can be formulated as a semi-definite program. Thus, it can be efficiently solved via numerical methods.

Lemma 1 (**Optimizing** α)**.** *Given subgraphs and their corresponding activation probabilities, optimizing the mixing matrix can be formulated as a semi-definite programming problem:*

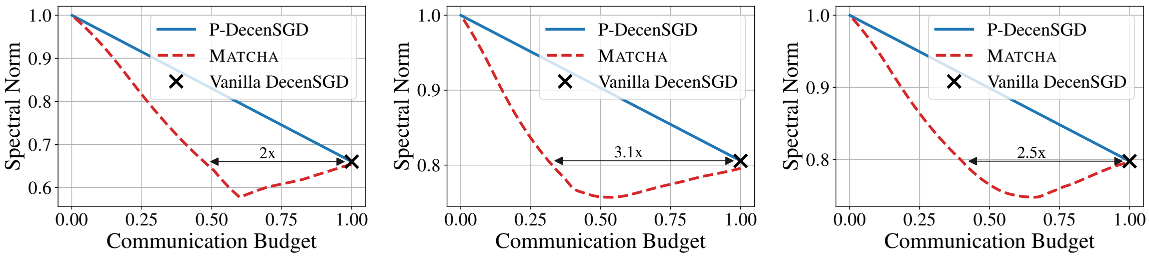
$$\min_{\rho,\alpha,\beta} \quad \rho, \quad \textit{subject to} \quad \alpha^2 - \beta \leq 0, \; \mathbf{I} - 2\alpha\overline{\mathbf{L}} + \beta[\overline{\mathbf{L}}^2 + 2\widetilde{\mathbf{L}}] - \frac{1}{m}\mathbf{1}\mathbf{1}^\top \preceq \rho\mathbf{I}, \tag{8}$$

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

Dependence on Communication Budget. In Figure 3, we present simulation results on how the minimal spectral norm ρ (solution of (8)) changes along with the communication budget. Recall that a lower spectral norm means better error-convergence in terms of iterations. It can be observed that Matcha can reduce 50%+ communication time while preserving the same spectral norm as vanilla DecenSGD. By setting a proper communication budget (for instance CB $\approx$ 0.4 in Figure 3b), 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. Moreover, even if one sets a very low communication budget, since the spectral norm only influences the higher order terms in (7), Matcha can still achieve the rate $\mathcal{O}(1/\sqrt{mK})$ after sufficiently large number of iterations. These theoretical findings are corroborated by extensive experiments in Section 5.

5 Experimental Results

Experimental Setting We evaluate the performance of the proposed algorithm in multiple deep learning tasks: (1) image classification on CIFAR-10 and CIFAR-100 [15]; (2) Language modeling on Penn Treebank corpus (PTB) dataset [19]. All training datasets are evenly partitioned over a network of workers. All algorithms are trained for sufficiently long times until convergence or overfitting. Besides, *in order to guarantee a fair comparison for each task, the learning rate is fine-tuned for*



(a) Graph in Figure 1: 8 nodes and maximal degree is 5.

(b) Geometric graph: 16 nodes and maximal degree is 10.

(c) Erdős-Rényi's graph: 16 nodes and maximal degree is 8.

Figure 3: Examples on how the spectral norm ρ varies over communication budget in MATCHA. Lower spectral norm means better error-convergence with respect to iterations.

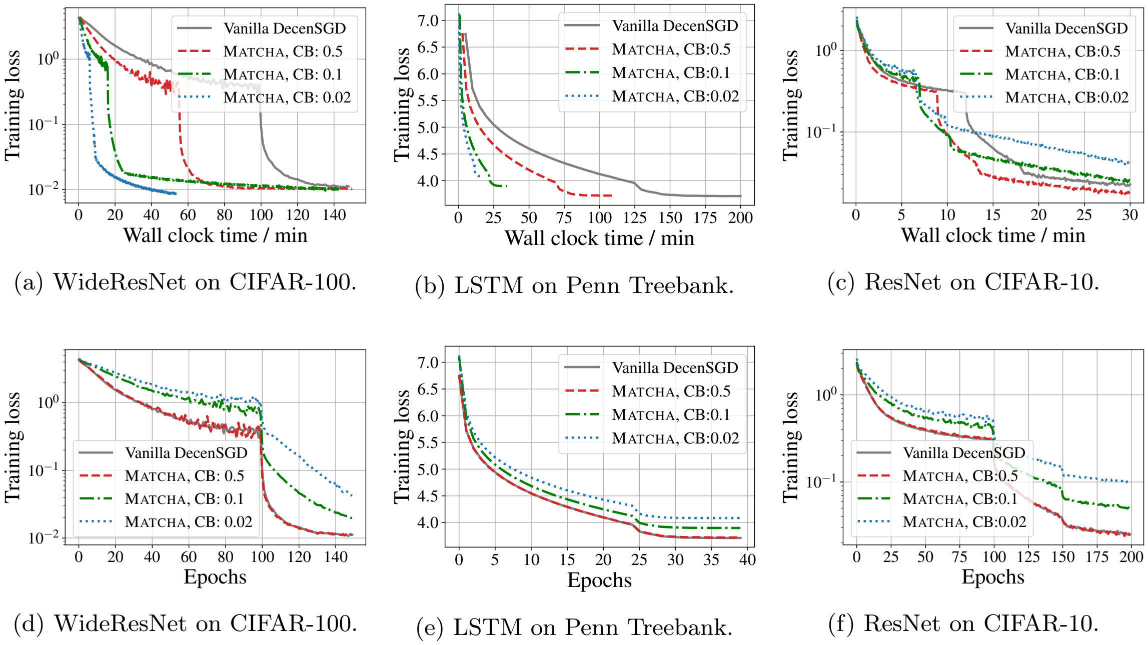
vanilla DecenSGD and kept the same in all other algorithms. More detailed descriptions on the datasets and training configurations are provided in Appendix A.1.

Effectiveness of MATCHA. We compare the performance of MATCHA with various communication budgets $(2\%, 10\%, 50\%)$ and vanilla DecenSGD in Figure 4. The base communication topology is shown in Figure 1. From Figures 4d to 4f, one can observe that when the communication budget is set to 50%, MATCHA has the nearly identical training loss as vanilla DecenSGD at every epoch. But it only requires, at most, half of the communication time per iteration. This empirical finding reinforces the claim regarding the similarity of the algorithms' performance in terms of epochs in Section 4 (see Figure 3a). When we continue to decrease the communication budget, MATCHA attains significantly faster convergence with respect to wall-clock time in communication-intense tasks. In particular, the proposed algorithm can reduce 98% communication time per iteration and achieve a training loss of 0.1 using 5x less time than vanilla DecenSGD on CIFAR-100 (see Figure 4a).

Effects of Base Communication Topology. In order to further verify the generality of MATCHA, we evaluate it on another base topology with varying connectivity using 16 worker nodes. In Figure 5, we present experimental results on three different base topologies, which are random geometric graphs and have different maximal degrees. In particular, when the maximal degree is 10 (see Figure 5b), MATCHA with communication budget 40% not only can reduce the communication time per iteration by $1/0.4 = 2.5$x, but also has lower error than vanilla DecenSGD. This result corroborates its corresponding spectral norm versus communication budget curve shown in Figure 3b. When we further increase the density of the base topology (see Figure 5c), MATCHA reduces communication time per iteration by $1/0.3 \simeq 3.3$x without hurting the error-convergence.

Another interesting observation is that MATCHA gives more 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 Sections 3 and 4, a naive way to reduce the communication time per iteration is to introduce a communication frequency for the whole



(a) WideResNet on CIFAR-100.

(b) LSTM on Penn Treebank.

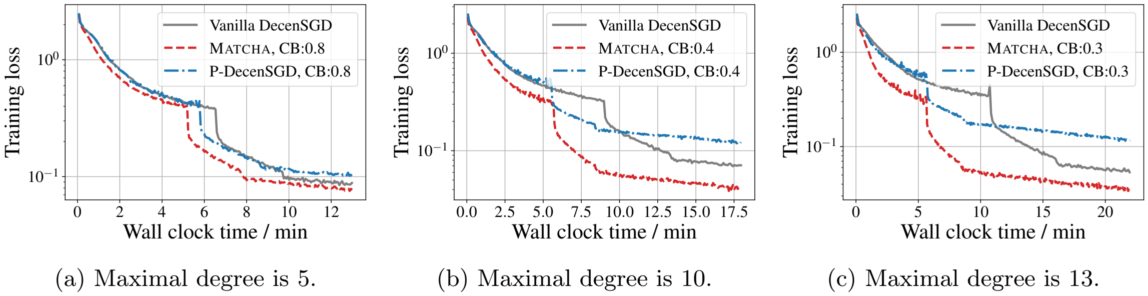
(c) ResNet on CIFAR-10.

(d) WideResNet on CIFAR-100.

(e) LSTM on Penn Treebank.

(f) ResNet on CIFAR-10.

Figure 4: Varying communication budget (CB) in Matcha.



(a) Maximal degree is 5.

(b) Maximal degree is 10.

(c) Maximal degree is 13.

Figure 5: ResNet-50 on CIFAR-10: **16 nodes on geometric graphs**.

base graph [35, 31]. Instead, in Matcha, we allow 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, Matcha consistently outperforms periodic DecenSGD. More results are presented in the Appendix.

6 Concluding Remarks

In this paper, we have proposed Matcha to reduce and control the communication delay of decentralized SGD algorithm in any general topology worker networks. The key idea in Matcha is that workers communicate over the connectivity-critical links with high priority, which we achieve via matching decomposition sampling. Rigorous theoretical analysis and experimental results show that MATCHA can reduce the communication delay while maintaining the same error-convergence rate in terms of epochs. Future directions includes adaptively changing the communication time per iteration as [34], and extending MATCHA to directed communication graphs.

References

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

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

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

[4] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2508–2530, 2006.

[5] Tianyi Chen, Georgios Giannakis, Tao Sun, and Wotao Yin. Lag: Lazily aggregated gradient for communication-efficient distributed learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060, 2018.

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

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

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Forrest N Iandola, Matthew W Moskewicz, Khalid Ashraf, and Kurt Keutzer. Firecaffe: near-linear acceleration of deep neural network training on compute clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2592–2600, 2016.

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

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

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

[13] Soummya Kar, José MF Moura, and Kavita Ramanan. Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication. *IEEE Transactions on Information Theory*, 58(6):3575–3605, 2012.

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

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

[16] Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *OSDI*, volume 14, pages 583–598, 2014.

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

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

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

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

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

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

[23] Reza Olfati-Saber. Algebraic connectivity ratio of Ramanujan graphs. In *2007 American Control Conference*, pages 4619–4624. IEEE, 2007.

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

[25] Anit Kumar Sahu, Dusan Jakovetic, Dragana Bajovic, and Soummya Kar. Communication-efficient distributed strongly convex stochastic optimization: Non-asymptotic rates. *arXiv preprint arXiv:1809.02920*, 2018.

[26] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Sparse binary compression: Towards distributed deep learning with minimal communication. *arXiv preprint arXiv:1805.08768*, 2018.

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

[28] Zebang Shen, Aryan Mokhtari, Tengfei Zhou, Peilin Zhao, and Hui Qian. Towards more efficient stochastic decentralized learning: Faster convergence and sparse communication. *arXiv preprint arXiv:1805.09969*, 2018.

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

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

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

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

[33] Hongyi Wang, Scott Sievert, Shengchao Liu, Zachary Charles, Dimitris Papailiopoulos, and Stephen Wright. Atomo: Communication-efficient learning via atomic sparsification. In *Advances in Neural Information Processing Systems*, pages 9850–9861, 2018.

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

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

[36] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient sparsification for communication-efficient distributed optimization. *arXiv preprint arXiv:1710.09854*, 2017.

[37] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. TernGrad: Ternary gradients to reduce communication in distributed deep learning. *arXiv preprint arXiv:1705.07878*, 2017.

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

[39] Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted SGD for non-convex optimization with faster convergence and less communication. *arXiv preprint arXiv:1807.06629*, 2018.

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

[41] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

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

[43] Hao Zhang, Zeyu Zheng, Shizhen Xu, Wei Dai, Qirong Ho, Xiaodan Liang, Zhiting Hu, Jinliang Wei, Pengtao Xie, and Eric P Xing. Poseidon: An efficient communication architecture for distributed deep learning on {GPU} clusters. In *2017 {USENIX} Annual Technical Conference ({USENIX}{ATC} 17)*, pages 181–193, 2017.

A More Experimental Results

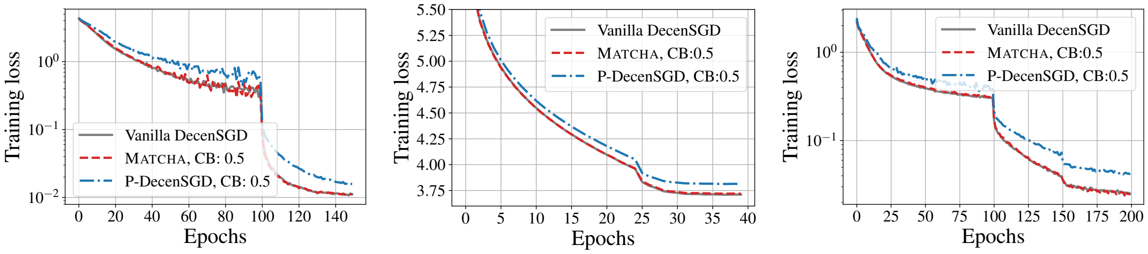
A.1 Detailed Experimental Setting

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. We train ResNet-50 [8], and WideResNet-28×10 [41] on the image classification tasks. A two-layer LSTM with 1500 hidden nodes in each layer [24] is adopted for language modeling. For the training on PTB dataset, 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. Unless otherwise stated, the training procedure is performed in a network of 8 nodes, each of which is equipped with one NVIDIA TitanX Maxwell GPU and has a 5000 MB/s Ethernet interface. Matcha is implemented with PyTorch and MPI4Py.

A.2 More Results

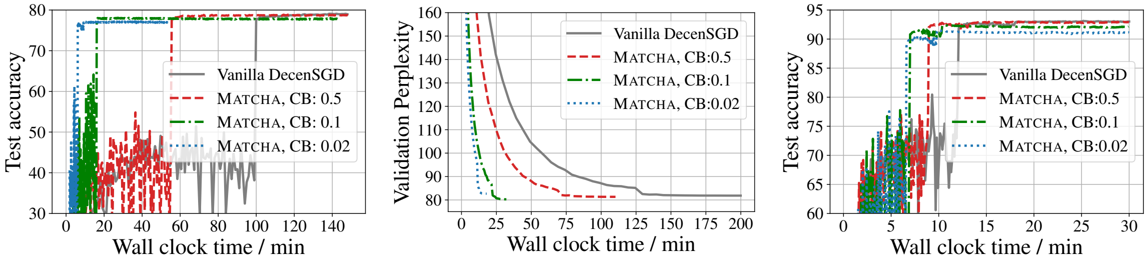


(a) WideResNet on CIFAR-100.

(b) LSTM on Penn Treebank.

(c) ResNet on CIFAR-10.

Figure 6: Comparision of Matcha and P-DecenSGD. While Matcha has nearly identical error-convergence to vanilla DecenSGD, P-DecenSGD performs consistently worse in all tasks.

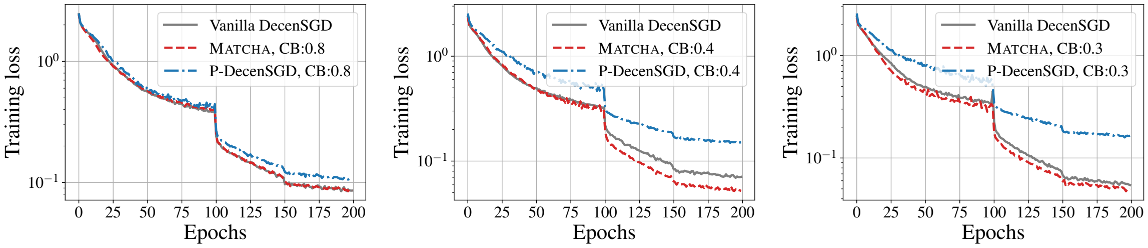


(a) WideResNet on CIFAR-100.

(b) LSTM on Penn Treebank.

(c) ResNet on CIFAR-10.

Figure 7: Test accuracy of Matcha on different training tasks.

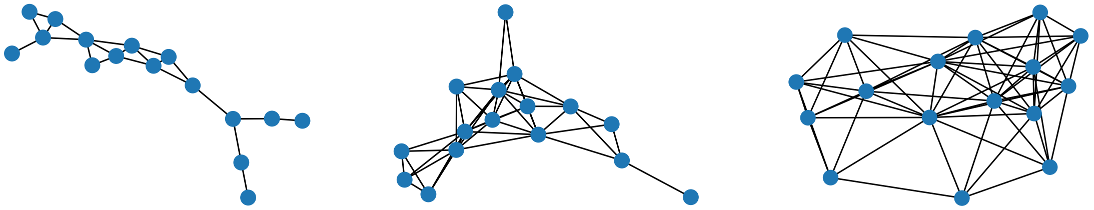


(a) Maximal degree is 5.

(b) Maximal degree is 10.

(c) Maximal degree is 13.

Figure 8: 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.

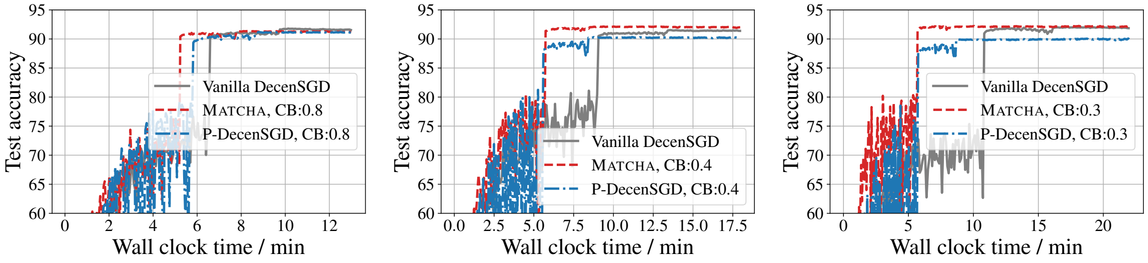


(a) Maximal degree is 5.

(b) Maximal degree is 10.

(c) Maximal degree is 13.

Figure 9: Different geometric topologies used in Figures 5 and 8.



(a) Maximal degree is 5.

(b) Maximal degree is 10.

(c) Maximal degree is 13.

Figure 10: Test accuracy of Matcha on different topologies with 16 nodes. Matcha can even have higher test accuracy than vanilla DecenSGD by setting a proper communication budget.

B Proofs of Theorem 1 and Corollary 1

B.1 Preliminaries

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

$$\mathbf{X}^{(k)} = \left[\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, \ldots, \mathbf{x}_m^{(k)}\right], \tag{9}$$

$$\mathbf{G}^{(k)} = \left[g_1(\mathbf{x}_1^{(k)}), g_2(\mathbf{x}_2^{(k)}), \ldots, g_m(\mathbf{x}_m^{(k)})\right], \tag{10}$$

$$\nabla\mathbf{F}^{(k)} = \left[\nabla F_1(\mathbf{x}_1^{(k)}), \nabla F_2(\mathbf{x}_2^{(k)}), \ldots, \nabla F_m(\mathbf{x}_m^{(k)})\right]. \tag{11}$$

Recall the assumptions we make:

$$\left\|\nabla F_i(\mathbf{x}) - \nabla F_i(\mathbf{y})\right\| \leq L \left\|\mathbf{x} - \mathbf{y}\right\|, \tag{12}$$

$$\mathbb{E}\left[g_i(\mathbf{x})|\mathbf{x}\right] = \nabla F_i(\mathbf{x}), \tag{13}$$

$$\mathbb{E}\left[\left\|g_i(\mathbf{x}) - \nabla F_i(\mathbf{x})\right\|^2 |\mathbf{x}\right] \leq \sigma^2. \tag{14}$$

B.2 Lemmas

Lemma 2. *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 $\mathbf{B} \in \mathbb{R}^{d\times m}$,*

$$\mathbb{E}\left[\left\|\mathbf{B}\left(\prod_{l=1}^{n}\mathbf{W}^{(l)} - \mathbf{J}\right)\right\|_F^2\right] \leq \rho^n \left\|\mathbf{B}\right\|_F^2 \tag{15}$$

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

$$\mathbf{A}_{1,n} = \prod_{k=1}^{n}\left(\mathbf{W}^{(k)} - \mathbf{J}\right) = \mathbf{A}_{1,n-1}\left(\mathbf{W}^{(n)} - \mathbf{J}\right). \tag{16}$$

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

$$\begin{aligned}
\mathbb{E}_{\mathbf{W}^{(n)}}\left[\left\|\mathbf{B}\mathbf{A}_{1,n}\right\|_{\mathrm{F}}^2\right] =& \sum_{i=1}^{d}\mathbb{E}_{\mathbf{W}^{(n)}}\left[\left\|\mathbf{b}_i^\top\mathbf{A}_{1,n}\right\|^2\right] & (17)\\
=& \sum_{i=1}^{d}\mathbb{E}_{\mathbf{W}^{(n)}}\left[\mathbf{b}_i^\top\mathbf{A}_{1,n-1}(\mathbf{W}^{(n)\top}\mathbf{W}^{(n)} - \mathbf{J})\mathbf{A}_{1,n-1}^\top\mathbf{b}_i\right] & (18)\\
=& \sum_{i=1}^{d}\mathbf{b}_i^\top\mathbf{A}_{1,n-1}\mathbb{E}_{\mathbf{W}^{(n)}}\left[(\mathbf{W}^{(n)\top}\mathbf{W}^{(n)} - \mathbf{J})\right]\mathbf{A}_{1,n-1}^\top\mathbf{b}_i. & (19)
\end{aligned}$$

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

$$\mathbb{E}_{\mathbf{W}^{(n)}}\left[\|\mathbf{B}\mathbf{A}_{1,n}\|_{\mathrm{F}}^2\right] = \sum_{i=1}^{d} \mathbf{v}_i^{\top}\mathbf{C}\mathbf{v}_i \tag{20}$$

$$\leq \sigma_{\max}(\mathbf{C}) \sum_{i=1}^{d} \mathbf{v}_i^{\top}\mathbf{v}_i \tag{21}$$

$$= \rho \left\|\mathbf{B}\mathbf{A}_{1,n-1}\right\|_{\mathrm{F}}^2 . \tag{22}$$

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

$$\mathbb{E}_{\mathbf{W}^{(1)}} \ldots \mathbb{E}_{\mathbf{W}^{(n-1)}} \mathbb{E}_{\mathbf{W}^{(n)}}\left[\|\mathbf{B}\mathbf{A}_{1,n}\|_{\mathrm{F}}^2\right] \leq \rho^n \left\|\mathbf{B}\right\|_{\mathrm{F}}^2 . \tag{23}$$

Here, we complete the proof. □

B.3 Proof of Theorem 1

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

$$F(\overline{\mathbf{x}}^{(k+1)}) - F(\overline{\mathbf{x}}^{(k)}) \leq \left\langle \nabla F(\overline{\mathbf{x}}^{(k)}),\, \overline{\mathbf{x}}^{(k+1)} - \overline{\mathbf{x}}^{(k)} \right\rangle + \frac{L}{2}\left\|\overline{\mathbf{x}}^{(k+1)} - \overline{\mathbf{x}}^{(k)}\right\|^2 . \tag{24}$$

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

$$F(\overline{\mathbf{x}}^{(k+1)}) - F(\overline{\mathbf{x}}^{(k)}) \leq -\eta \left\langle \nabla F(\overline{\mathbf{x}}^{(k)}),\, \frac{\mathbf{G}^{(k)}\mathbf{1}}{m} \right\rangle + \frac{\eta^2 L}{2}\left\|\frac{\mathbf{G}^{(k)}\mathbf{1}}{m}\right\|^2 . \tag{25}$$

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

$$\mathbb{E}_k\left[F(\overline{\mathbf{x}}^{(k+1)})\right] - F(\overline{\mathbf{x}}^{(k)}) \leq -\eta \left\langle \nabla F(\overline{\mathbf{x}}^{(k)}),\, \frac{\nabla\mathbf{F}^{(k)}\mathbf{1}}{m} \right\rangle + \frac{\eta^2 L}{2}\mathbb{E}_k\left[\left\|\frac{\mathbf{G}^{(k)}\mathbf{1}}{m}\right\|^2\right] . \tag{26}$$

For the first term in (26), since $2\langle a,\, b\rangle = \|a\|^2 + \|b\|^2 - \|a-b\|^2$, we have

$$\left\langle \nabla F(\overline{\mathbf{x}}^{(k)}),\, \frac{\nabla\mathbf{F}^{(k)}\mathbf{1}}{m} \right\rangle = \left\langle \nabla F(\overline{\mathbf{x}}^{(k)}),\, \frac{1}{m}\sum_{i=1}^{m}\nabla F_i(\mathbf{x}_i^{(k)}) \right\rangle \tag{27}$$

$$= \frac{1}{2}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2 + \left\|\frac{1}{m}\sum_{i=1}^{m}\nabla F_i(\mathbf{x}_i^{(k)})\right\|^2 - \left\|\nabla F(\overline{\mathbf{x}}^{(k)}) - \frac{1}{m}\sum_{i=1}^{m}\nabla F_i(\mathbf{x}_i^{(k)})\right\|^2\right] \tag{28}$$

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

$$\left\|\nabla F(\overline{\mathbf{x}}^{(k)}) - \frac{1}{m}\sum_{i=1}^{m}\nabla F_i(\mathbf{x}_i^{(k)})\right\|^2 = \left\|\frac{1}{m}\sum_{i=1}^{m}\left[\nabla F_i(\overline{\mathbf{x}}^{(k)}) - \nabla F_i(\mathbf{x}_i^{(k)})\right]\right\|^2 \tag{29}$$

$$\overset{\text{Jensen's Inequality}}{\leq} \frac{1}{m}\sum_{i=1}^{m}\left\|\nabla F_i(\overline{\mathbf{x}}^{(k)}) - \nabla F_i(\mathbf{x}_i^{(k)})\right\|^2 \tag{30}$$

$$\leq \frac{L^2}{m}\sum_{i=1}^{m}\left\|\overline{\mathbf{x}}^{(k)} - \mathbf{x}_i^{(k)}\right\|^2 \tag{31}$$

where the last inequality follows the Lipschitz smooth assumption. Then, plugging (31) into (28), we obtain

$$\left\langle \nabla F(\overline{\mathbf{x}}^{(k)}),\ \frac{\nabla \mathbf{F}^{(k)}\mathbf{1}}{m} \right\rangle \geq \frac{1}{2}\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2 + \frac{1}{2}\left\|\frac{\nabla \mathbf{F}^{(k)}\mathbf{1}}{m}\right\|^2 - \frac{L^2}{2m}\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^2. \tag{32}$$

Next, for the second part in (26),

$$\begin{aligned}
\mathbb{E}_k\left[\left\|\frac{1}{m}\sum_{i=1}^{m} g_i(\mathbf{x}_i^{(k)})\right\|^2\right] =& \mathbb{E}_k\left[\frac{1}{m}\sum_{i=1}^{m}\left[g_i(\mathbf{x}_i^{(k)}) - \nabla F_i(\mathbf{x}_i^{(k)}) + \nabla F_i(\mathbf{x}_i^{(k)})\right]\right]^2 && (33)\\
=& \frac{1}{m^2}\sum_{i=1}^{m}\mathbb{E}_k\left[\left\|g_i(\mathbf{x}_i^{(k)}) - \nabla F_i(\mathbf{x}_i^{(k)})\right\|^2\right] + \left\|\frac{1}{m}\sum_{i=1}^{m}\nabla F_i(\mathbf{x}_i^{(k)})\right\|^2 && (34)\\
\leq& \frac{\sigma^2}{m} + \left\|\frac{\nabla \mathbf{F}^{(k)}\mathbf{1}}{m}\right\|^2 && (35)
\end{aligned}$$

where the last inequality is according to the bounded variance assumption. Then, combining (32) and (35) and taking the total expectation over all random variables, one can obtain:

$$\begin{aligned}
\mathbb{E}\left[F(\overline{\mathbf{x}}^{(k+1)}) - F(\overline{\mathbf{x}}^{(k)})\right] \leq& -\frac{\eta}{2}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right] - \frac{\eta}{2}(1-\eta L)\mathbb{E}\left[\left\|\frac{\nabla \mathbf{F}^{(k)}\mathbf{1}}{m}\right\|^2\right] + \\
& \frac{\eta L^2}{2m}\mathbb{E}\left[\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^2\right] + \frac{\eta^2 L\sigma^2}{2m}. && (36)
\end{aligned}$$

Summing over all iterates and taking the average,

$$\begin{aligned}
\frac{\mathbb{E}\left[F(\overline{\mathbf{x}}^{K}) - F(\overline{\mathbf{x}}^{(1)})\right]}{K} \leq& -\frac{\eta}{2}\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right] - \frac{\eta}{2}(1-\eta L)\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\frac{\nabla \mathbf{F}^{(k)}\mathbf{1}}{m}\right\|^2\right] + \\
& \frac{\eta L^2}{2mK}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^2\right] + \frac{\eta^2 L\sigma^2}{2m}. && (37)
\end{aligned}$$

By minor rearranging, we get

$$\begin{aligned}
\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right] \leq& \frac{2\mathbb{E}\left[F(\overline{\mathbf{x}}^{(1)}) - F(\overline{\mathbf{x}}^{(K)})\right]}{\eta K} - \frac{1-\eta L}{m}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\frac{\nabla \mathbf{F}^{(k)}\mathbf{1}}{m}\right\|^2\right] + \\
& \frac{L^2}{mK}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^2\right] + \frac{\eta L\sigma^2}{m} && (38)\\
\leq& \frac{2\left[F(\overline{\mathbf{x}}^{(1)}) - F_{\inf}\right]}{\eta K} - \frac{1-\eta L}{m}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\frac{\nabla \mathbf{F}^{(k)}\mathbf{1}}{m}\right\|^2\right] + \\
& \frac{L^2}{mK}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^2\right] + \frac{\eta L\sigma^2}{m}. && (39)
\end{aligned}$$

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

$$\begin{aligned}\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J}) =& \left(\mathbf{X}^{(k-1)} - \eta\mathbf{G}^{(k-1)}\right)\mathbf{W}^{(k-1)}(\mathbf{I}-\mathbf{J}) && (40)\\ =&\mathbf{X}^{(k-1)}(\mathbf{I}-\mathbf{J})\mathbf{W}^{(k-1)} - \eta\mathbf{G}^{(k-1)}\mathbf{W}^{(k-1)}(\mathbf{I}-\mathbf{J}) && (41)\\ &\vdots && (42)\\ =&\mathbf{X}^{(1)}(\mathbf{I}-\mathbf{J})\prod_{q=1}^{k-1}\mathbf{W}^{(q)} - \eta\sum_{q=1}^{k-1}\mathbf{G}^{(q)}\left(\prod_{l=q}^{k-1}\mathbf{W}^{(l)} - \mathbf{J}\right). && (43)\end{aligned}$$

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

$$\begin{aligned}\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^2 =& \eta^2\left\|\sum_{q=1}^{k-1}\mathbf{G}^{(q)}\left(\prod_{l=q}^{k-1}\mathbf{W}^{(l)} - \mathbf{J}\right)\right\|_{\mathrm{F}}^2 && (44)\\ =&\eta^2\left\|\sum_{q=1}^{k-1}\left(\mathbf{G}^{(q)} - \nabla\mathbf{F}^{(q)} + \nabla\mathbf{F}^{(q)}\right)\left(\prod_{l=q}^{k-1}\mathbf{W}^{(l)} - \mathbf{J}\right)\right\|_{\mathrm{F}}^2 && (45)\\ \leq& 2\eta^2\underbrace{\left\|\sum_{q=1}^{k-1}\left(\mathbf{G}^{(q)} - \nabla\mathbf{F}^{(q)}\right)\left(\prod_{l=q}^{k-1}\mathbf{W}^{(l)} - \mathbf{J}\right)\right\|_{\mathrm{F}}^2}_{T_1} + 2\eta^2\underbrace{\left\|\sum_{q=1}^{k-1}\nabla\mathbf{F}^{(q)}\left(\prod_{l=q}^{k-1}\mathbf{W}^{(l)} - \mathbf{J}\right)\right\|_{\mathrm{F}}^2}_{T_2}. && (46)\end{aligned}$$

For the first term T_1 in (46), we have

$$\begin{aligned}\mathbb{E}\left[T_1\right] =& \sum_{q=1}^{k-1}\mathbb{E}\left[\left\|\left(\mathbf{G}^{(q)} - \nabla\mathbf{F}^{(q)}\right)\left(\prod_{l=q}^{k-1}\mathbf{W}^{(l)} - \mathbf{J}\right)\right\|_{\mathrm{F}}^2\right] && (47)\\ \leq& \sum_{q=1}^{k-1}\rho^{k-q}\mathbb{E}\left[\left\|\mathbf{G}^{(q)} - \nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] && (48)\\ \leq& m\sigma^2\rho\left(1+\rho+\rho^2+\cdots+\rho^{k-2}\right) && (49)\\ \leq& \frac{m\sigma^2\rho}{1-\rho} && (50)\end{aligned}$$

where (48) comes from Lemma 2. For the second term T_2 in (46), define $\mathbf{A}_{q,p} = \prod_{l=q}^{p}\mathbf{W}^{(l)} - \mathbf{J}$. Then,

$$\begin{aligned}\mathbb{E}\left[T_2\right] =& \sum_{q=1}^{k-1}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\mathbf{A}_{q,k-1}\right\|_{\mathrm{F}}^2\right] + \sum_{q=1}^{k-1}\sum_{p=1,p\neq q}^{k-1}\mathbb{E}\left[\mathrm{Tr}\{\mathbf{A}_{q,k-1}^\top\nabla\mathbf{F}^{(q)\top}\nabla\mathbf{F}^{(p)}\mathbf{A}_{p,k-1}\}\right] && (51)\\ \leq& \sum_{q=1}^{k-1}\rho^{k-q}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] + \sum_{q=1}^{k-1}\sum_{p=1,p\neq q}^{k-1}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\mathbf{A}_{q,k-1}\right\|_{\mathrm{F}}\left\|\nabla\mathbf{F}^{(p)}\mathbf{A}_{p,k-1}\right\|_{\mathrm{F}}\right] && (52)\\ \leq& \sum_{q=1}^{k-1}\rho^{k-q}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] + \sum_{q=1}^{k-1}\sum_{p=1,p\neq q}^{k-1}\mathbb{E}\left[\frac{1}{2\epsilon}\left\|\nabla\mathbf{F}^{(q)}\mathbf{A}_{q,k-1}\right\|_{\mathrm{F}}^2 + \frac{\epsilon}{2}\left\|\nabla\mathbf{F}^{(p)}\mathbf{A}_{p,k-1}\right\|_{\mathrm{F}}^2\right] && (53)\\ \leq& \sum_{q=1}^{k-1}\rho^{k-q}\mathbb{E}\left[\left\|\nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] + \sum_{q=1}^{k-1}\sum_{p=1,p\neq q}^{k-1}\mathbb{E}\left[\frac{\rho^{k-q}}{2\epsilon}\left\|\nabla\mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2 + \frac{\rho^{k-p}\epsilon}{2}\left\|\nabla\mathbf{F}^{(p)}\right\|_{\mathrm{F}}^2\right] && (54)\end{aligned}$$

where (53) follows Young's Inequality: $2ab \leq a^2/\epsilon + \epsilon b^2, \forall \epsilon > 0$ and (54) follows Lemma 2. Set $\epsilon = \rho^{\frac{p-q}{2}}$, then we have

$$\begin{aligned}
\mathbb{E}\left[T_2\right] \leq & \sum_{q=1}^{k-1} \rho^{k-q} \mathbb{E}\left[\left\|\nabla \mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] + \frac{1}{2} \sum_{q=1}^{k-1} \sum_{p=1, p\neq q}^{k-1} \sqrt{\rho}^{2k-p-q} \cdot \mathbb{E}\left[\left\|\nabla \mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2 + \left\|\nabla \mathbf{F}^{(p)}\right\|_{\mathrm{F}}^2\right] && (55)\\
= & \sum_{q=1}^{k-1} \rho^{k-q} \mathbb{E}\left[\left\|\nabla \mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] + \sum_{q=1}^{k-1} \left[\sqrt{\rho}^{k-q} \mathbb{E}\left[\left\|\nabla \mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] \cdot \sum_{p=1, p\neq q}^{k-1} \sqrt{\rho}^{k-p}\right] && (56)\\
= & \sum_{q=1}^{k-1} \rho^{k-q} \mathbb{E}\left[\left\|\nabla \mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] + \sum_{q=1}^{k-1} \left[\sqrt{\rho}^{k-q} \mathbb{E}\left[\left\|\nabla \mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] \cdot \left(\sum_{p=1}^{k-1} \sqrt{\rho}^{k-p} - \sqrt{\rho}^{k-q}\right)\right] && (57)\\
\leq & \frac{\sqrt{\rho}}{1-\sqrt{\rho}} \sum_{q=1}^{k-1} \sqrt{\rho}^{k-q} \mathbb{E}\left[\left\|\nabla \mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right]. && (58)
\end{aligned}$$

Combining (50) and (58) together,

$$\begin{aligned}
\frac{1}{mK} \sum_{i=1}^{K} \mathbb{E}\left[\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^2\right] \leq & \frac{2\eta^2\sigma^2\rho}{1-\rho} + \frac{2\eta^2}{m} \frac{\sqrt{\rho}}{1-\sqrt{\rho}} \frac{1}{K} \sum_{k=1}^{K} \sum_{q=1}^{k-1} \sqrt{\rho}^{k-q} \mathbb{E}\left[\left\|\nabla \mathbf{F}^{(q)}\right\|_{\mathrm{F}}^2\right] && (59)\\
= & \frac{2\eta^2\sigma^2\rho}{1-\rho} + \frac{2\eta^2}{m} \frac{\sqrt{\rho}}{1-\sqrt{\rho}} \frac{1}{K} \sum_{k=1}^{K} \left[\mathbb{E}\left[\left\|\nabla \mathbf{F}^{(k)}\right\|_{\mathrm{F}}^2\right] \sum_{q=1}^{K-k} \sqrt{\rho}^{q}\right] && (60)\\
\leq & \frac{2\eta^2\sigma^2\rho}{1-\rho} + \frac{2\eta^2}{m} \frac{\sqrt{\rho}}{1-\sqrt{\rho}} \frac{1}{K} \sum_{k=1}^{K} \left[\mathbb{E}\left[\left\|\nabla \mathbf{F}^{(k)}\right\|_{\mathrm{F}}^2\right] \frac{\sqrt{\rho}}{1-\sqrt{\rho}}\right] && (61)\\
= & \frac{2\eta^2\sigma^2\rho}{1-\rho} + \frac{2\eta^2}{m} \frac{\rho}{(1-\sqrt{\rho})^2} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\left[\left\|\nabla \mathbf{F}^{(k)}\right\|_{\mathrm{F}}^2\right] && (62)
\end{aligned}$$

Plugging (62) back into (39), we have

$$\begin{aligned}
\frac{1}{K} \sum_{i=1}^{K} \mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right] \leq & \frac{2[F(\overline{\mathbf{x}}^{(1)}) - F_{\mathrm{inf}}]}{\eta K} - \frac{1-\eta L}{m} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\left[\left\|\frac{\nabla \mathbf{F}^{(k)}\mathbf{1}}{m}\right\|^2\right] + \frac{\eta L \sigma^2}{m} + \\
& \eta^2 L^2 \sigma^2 \frac{2\rho}{1-\rho} + \frac{\eta^2 L^2}{m} \frac{2\rho}{(1-\sqrt{\rho})^2} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\left[\left\|\nabla \mathbf{F}^{(k)}\right\|_{\mathrm{F}}^2\right] && (63)\\
\leq & \frac{2[F(\overline{\mathbf{x}}^{(1)}) - F_{\mathrm{inf}}]}{\eta K} + \frac{\eta L \sigma^2}{m} + \eta^2 L^2 \sigma^2 \frac{2\rho}{1-\rho} + \\
& \frac{\eta^2 L^2}{m} \frac{2\rho}{(1-\sqrt{\rho})^2} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\left[\left\|\nabla \mathbf{F}^{(k)}\right\|_{\mathrm{F}}^2\right] && (64)
\end{aligned}$$

where the last inequality comes from the fact that $\eta L \leq 1$. Here, we complete the proof.

B.4 Proof of Corollary 1

If we further assume the uniform boundedness of the gradients, i.e., $\left\|\nabla F_i(\mathbf{x}_i)\right\|^2 \leq D$, then we obtain

$$\frac{1}{K} \sum_{i=1}^{K} \mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right] \leq \frac{2[F(\overline{\mathbf{x}}^{(1)}) - F_{\mathrm{inf}}]}{\eta K} + \frac{\eta L \sigma^2}{m} + 2\eta^2 L^2 \rho \left(\frac{\sigma^2}{1-\rho} + \frac{D}{(1-\sqrt{\rho})^2}\right). \qquad (65)$$

When $\eta = \frac{1}{L}\sqrt{\frac{m}{K}}$, it follows that

$$\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right] \leq \frac{2L[F(\overline{\mathbf{x}}^{(1)})] - F_{\text{inf}}]}{\sqrt{mK}} + \frac{\sigma^2}{\sqrt{mK}} + \frac{2m\rho}{K}\left(\frac{\sigma^2}{1-\rho} + \frac{D}{(1-\sqrt{\rho})^2}\right). \tag{66}$$

B.5 Another Version of Corollary 1 with Weaker Assumption

Now let us assume

$$\|\nabla F_i(\mathbf{x}) - \nabla F(\mathbf{x})\| \leq \zeta^2. \tag{67}$$

Recall the inequality (62),

$$\frac{1}{mK}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\text{F}}^2\right] \leq \frac{2\eta^2\sigma^2\rho}{1-\rho} + \frac{2\eta^2}{m}\frac{\rho}{(1-\sqrt{\rho})^2}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla \mathbf{F}^{(k)}\right\|_{\text{F}}^2\right]. \tag{68}$$

Note that

$$\begin{aligned}
\left\|\nabla \mathbf{F}^{(k)}\right\|_{\text{F}}^2 =& \sum_{i=1}^{m}\left\|\nabla F_i(\mathbf{x}_i^{(k)})\right\|^2 & (69)\\
=& \sum_{i=1}^{m}\left\|\nabla F_i(\mathbf{x}_i^{(k)}) - \nabla F(\mathbf{x}_i^{(k)}) + \nabla F(\mathbf{x}_i^{(k)}) - \nabla F(\overline{\mathbf{x}}^{(k)}) + \nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2 & (70)\\
\leq& 3\sum_{i=1}^{m}\left[\left\|\nabla F_i(\mathbf{x}_i^{(k)}) - \nabla F(\mathbf{x}_i^{(k)})\right\|^2 + \left\|\nabla F(\mathbf{x}_i^{(k)}) - \nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2 + \left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right] & (71)\\
\leq& 3m\zeta^2 + 3L^2\sum_{i=1}^{m}\left\|\mathbf{x}_i^{(k)} - \overline{\mathbf{x}}^{(k)}\right\|^2 + 3m\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2 & (72)\\
=& 3m\zeta^2 + 3L^2\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\text{F}}^2 + 3m\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2. & (73)
\end{aligned}$$

Plugging (73) back into (68), we have

$$\begin{aligned}
\frac{1}{mK}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\text{F}}^2\right] \leq& \frac{2\eta^2\sigma^2\rho}{1-\rho} + \frac{6\eta^2\zeta^2\rho}{(1-\sqrt{\rho})^2} + \frac{6\eta^2L^2\rho}{(1-\sqrt{\rho})^2}\frac{1}{mK}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\text{F}}^2\right] + \\
&\frac{6\eta^2\rho}{(1-\sqrt{\rho})^2}\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right]. \qquad (74)
\end{aligned}$$

After minor rearranging, we get

$$\frac{1}{mK}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\mathbf{X}^{(k)}(\mathbf{I}-\mathbf{J})\right\|_{\text{F}}^2\right] \leq \frac{1}{1-C_1}\left[\frac{2\eta^2\sigma^2\rho}{1-\rho} + \frac{6\eta^2\zeta^2\rho}{(1-\sqrt{\rho})^2} + \frac{6\eta^2\rho}{(1-\sqrt{\rho})^2}\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right]\right] \tag{75}$$

where $C_1 = \frac{6\eta^2L^2\rho}{(1-\sqrt{\rho})^2}$. Here, the hyper-parameters should satisfy

$$C_1 = \frac{6\eta^2L^2\rho}{(1-\sqrt{\rho})^2} < \frac{1}{2}. \tag{76}$$

Then, plugging (75) back into (39), we have

$$\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right] \leq \frac{2[F(\overline{\mathbf{x}}^{(1)}) - F_{\text{inf}}]}{\eta K} + \frac{\eta L\sigma^2}{m} + \frac{1}{1-C_1}\frac{2\eta^2L^2\sigma^2\rho}{1-\rho} + \frac{C_1\zeta^2}{1-C_1} + \frac{C_1}{1-C_1}\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right]. \tag{77}$$

It follows that

$$\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{(k)})\right\|^2\right] \leq \left(\frac{2[F(\overline{\mathbf{x}}^{(1)}) - F_{\text{inf}}]}{\eta K} + \frac{\eta L\sigma^2}{m}\right)\frac{1-C_1}{1-2C_1} + \left(\frac{2\eta^2L^2\sigma^2\rho}{1-\rho} + \frac{6\eta^2L^2\zeta^2\rho}{(1-\sqrt{\rho})^2}\right)\frac{1}{1-2C_1} \tag{78}$$

$$= \left(\frac{2[F(\overline{\mathbf{x}}^{(1)}) - F_{\text{inf}}]}{\eta K} + \frac{\eta L\sigma^2}{m}\right)D_1 + 2\eta^2L^2D_2\left(\frac{\sigma^2}{1-\rho} + \frac{3\zeta^2\rho}{(1-\sqrt{\rho})^2}\right) \tag{79}$$

where $D_1 = (1-C_1)/(1-2C_1)$ and $D_2 = 1/(1-2C_1)$. Compared to (66), this new version with weaker assumption does not provide any new insights. However, it increases the complexity of the error bound.

C Proof of Theorem 2 and Lemma 1

C.1 Proof of Theorem 2

The proof contains two 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 $\rho < 1$.
Recall that $\{p_j\}_{j=1}^{M}$ is the solution of convex optimization problem (4). Let $p_0 = \text{CB}$, then we have

$$\lambda_2(\sum_{j=1}^{M} p_j\mathbf{L}_j) \geq \lambda_2(p_0\sum_{j=1}^{M}\mathbf{L}_j) = p_0\lambda_2(\sum_{j=1}^{M}\mathbf{L}_j) > 0. \tag{80}$$

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

$$\left\|\mathbb{E}\left[\mathbf{W}^{(k)\top}\mathbf{W}^{(k)}\right] - \mathbf{J}\right\|_2 = \left\|\mathbb{E}\left[\left(\mathbf{I} - \alpha\mathbf{L}^{(k)}\right)^\top\left(\mathbf{I} - \alpha\mathbf{L}^{(k)}\right)\right] - \mathbf{J}\right\|_2 \tag{81}$$

$$= \left\|\mathbf{I} - 2\alpha\mathbb{E}\left[\mathbf{L}^{(k)}\right] + \alpha^2\mathbb{E}\left[\mathbf{L}^{(k)\top}\mathbf{L}^{(k)}\right] - \mathbf{J}\right\|_2 \tag{82}$$

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,

$$\mathbb{E}\left[\mathbf{L}^{(k)}\right] = \sum_{j=1}^{M} p_j \mathbf{L}_j \tag{83}$$

$$\mathbb{E}\left[\mathbf{L}^{(k)\top}\mathbf{L}^{(k)}\right] = \sum_{j=1}^{M} p_j^2 \mathbf{L}_j^2 + \sum_{j=1}^{M} \sum_{t=1, t\neq j}^{M} p_j p_t \mathbf{L}_j^\top \mathbf{L}_t + \sum_{j=1}^{M} p_j(1-p_j)\mathbf{L}_j^2 \tag{84}$$

$$= \left(\sum_{j=1}^{M} p_j \mathbf{L}_j\right)^2 + \sum_{j=1}^{M} p_j(1-p_j)\mathbf{L}_j^2 \tag{85}$$

$$= \left(\sum_{j=1}^{M} p_j \mathbf{L}_j\right)^2 + 2\sum_{j=1}^{M} p_j(1-p_j)\mathbf{L}_j. \tag{86}$$

Plugging (83) and (86) back into (82), we get

$$\left\|\mathbb{E}\left[\mathbf{W}^{(k)\top}\mathbf{W}^{(k)}\right] - \mathbf{J}\right\|_2 = \left\|\left(\mathbf{I} - \alpha\sum_{j=1}^{M} p_j \mathbf{L}_j\right)^2 + 2\alpha^2 \sum_{j=1}^{M} p_j(1-p_j)\mathbf{L}_j - \mathbf{J}\right\|_2 \tag{87}$$

$$\leq \left\|\left(\mathbf{I} - \alpha\sum_{j=1}^{M} p_j \mathbf{L}_j\right)^2 - \mathbf{J}\right\|_2 + 2\alpha^2 \left\|\sum_{j=1}^{M} p_j(1-p_j)\mathbf{L}_j\right\|_2 \tag{88}$$

$$= \max\{|1-\alpha\lambda_2|^2, |1-\alpha\lambda_m|^2\} + 2\alpha^2\zeta \tag{89}$$

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

$$\frac{\partial h}{\partial \alpha} = -2\lambda(1-\alpha\lambda) + 4\alpha\zeta, \tag{90}$$

$$\frac{\partial^2 h}{\partial \alpha^2} = 2\lambda^2 + 4\zeta > 0. \tag{91}$$

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

$$\alpha^* = \frac{\lambda}{\lambda^2 + 2\zeta}, \tag{92}$$

$$h_\lambda(\alpha^*) = \frac{4\zeta^2}{(\lambda^2+2\zeta)^2} + \frac{2\lambda^2\zeta}{(\lambda^2+2\zeta)^2} = \frac{2\zeta}{\lambda^2+2\zeta}. \tag{93}$$

Furthermore, note that $h(0) = 1$ and $h_\lambda(\alpha)$ is a quadratic function. Since we prove that $\lambda_2 > 0$ (i.e., $\alpha^* > 0$), 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

$$\left\|\mathbb{E}\left[\mathbf{W}^{(k)\top}\mathbf{W}^{(k)}\right] - \mathbf{J}\right\|_2 \leq \max\{h_{\lambda_2}(\alpha), h_{\lambda_m}(\alpha)\} < 1. \tag{94}$$

C.2 Proof of Lemma 1

In the proof of Theorem 2, we have shown that the spectral norm ρ can be expanded as

$$\left\|\mathbb{E}\left[\mathbf{W}^{(k)\top}\mathbf{W}^{(k)}\right]-\mathbf{J}\right\|_2=\left\|\mathbf{I}-2\alpha\sum_{j=1}^{M}p_j\mathbf{L}_j+\alpha^2\left(\sum_{j=1}^{M}p_j\mathbf{L}_j\right)^2+2\alpha^2\sum_{j=1}^{M}p_j(1-p_j)\mathbf{L}_j-\mathbf{J}\right\|_2 \tag{95}$$

$$=\left\|\mathbf{I}-2\alpha\overline{\mathbf{L}}+\alpha^2\overline{\mathbf{L}}^2+2\alpha^2\widetilde{\mathbf{L}}-\mathbf{J}\right\|_2 \tag{96}$$

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:

$$\min_{\alpha}\quad \left\|\mathbf{I}-2\alpha\overline{\mathbf{L}}+\alpha^2[\overline{\mathbf{L}}^2+2\widetilde{\mathbf{L}}]-\mathbf{J}\right\|_2 \tag{97}$$

which is equivalent to

$$\begin{aligned}\min_{\rho,\alpha}\quad & \rho\\ \text{subject to}\quad & \mathbf{I}-2\alpha\overline{\mathbf{L}}+\alpha^2[\overline{\mathbf{L}}^2+2\widetilde{\mathbf{L}}]-\frac{1}{m}\mathbf{1}\mathbf{1}^\top\preceq\rho\mathbf{I}.\end{aligned} \tag{98}$$

However, directly solving (98) 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:

$$\begin{aligned}\min_{\rho,\alpha,\beta}\quad & \rho\\ \text{subject to}\quad & \alpha^2-\beta\leq 0,\ \mathbf{I}-2\alpha\overline{\mathbf{L}}+\beta[\overline{\mathbf{L}}^2+2\widetilde{\mathbf{L}}]-\frac{1}{m}\mathbf{1}\mathbf{1}^\top\preceq\rho\mathbf{I}.\end{aligned} \tag{99}$$

Now the constraints become linear matrix inequality constraints and (99) is the standard form of semi-definite programming. However, we need to further show that the solution of (99) is same as (98). We will prove this by contradiction. Suppose α_+,β_+,ρ_+ are the solution of problem (99) 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

$$\mathbf{I}-2\alpha_+\overline{\mathbf{L}}+(\alpha_+^2+c)[\overline{\mathbf{L}}^2+2\widetilde{\mathbf{L}}]-\frac{1}{m}\mathbf{1}\mathbf{1}^\top\preceq\rho_+\mathbf{I}. \tag{100}$$

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

$$\mathbf{I}-2\alpha_+\overline{\mathbf{L}}+\alpha_+^2[\overline{\mathbf{L}}^2+2\widetilde{\mathbf{L}}]-\frac{1}{m}\mathbf{1}\mathbf{1}^\top\prec\mathbf{I}-2\alpha_+\overline{\mathbf{L}}+(\alpha_+^2+c)[\overline{\mathbf{L}}^2+2\widetilde{\mathbf{L}}]-\frac{1}{m}\mathbf{1}\mathbf{1}^\top\preceq\rho_+\mathbf{I}. \tag{101}$$

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

$$\mathbf{I}-2\alpha_+\overline{\mathbf{L}}+\alpha_+^2[\overline{\mathbf{L}}^2+2\widetilde{\mathbf{L}}]-\frac{1}{m}\mathbf{1}\mathbf{1}^\top\preceq\rho_*\mathbf{I}\prec\rho_+\mathbf{I}. \tag{102}$$

So our assumption $\alpha_+^2<\beta_+$ cannot hold. The solutions of (99) must satisfy $\alpha^2=\beta$.

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

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

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 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 $\lambda_2(\mathbf{L}) > 0$