Beyond spectral gap: The role of the topology in decentralized learning

Thijs Vogels[*]
EPFL

Hadrien Hendrikx[*]
EPFL

Martin Jaggi
EPFL

Abstract

In data-parallel optimization of machine learning models, workers collaborate to improve their estimates of the model: more accurate gradients allow them to use larger learning rates and optimize faster. We consider the setting in which all workers sample from the same dataset, and communicate over a sparse graph (decentralized). In this setting, current theory fails to capture important aspects of real-world behavior. First, the 'spectral gap' of the communication graph is not predictive of its empirical performance in (deep) learning. Second, current theory does not explain that collaboration enables *larger* learning rates than training alone. In fact, it prescribes *smaller* learning rates, which further decrease as graphs become larger, failing to explain convergence in infinite graphs. This paper aims to paint an accurate picture of sparsely-connected distributed optimization when workers share the same data distribution. We quantify how the graph topology influences convergence in a quadratic toy problem and provide theoretical results for general smooth and (strongly) convex objectives. Our theory matches empirical observations in deep learning, and accurately describes the relative merits of different graph topologies. Code: `github.com/epfml/topology-in-decentralized-learning`

1 Introduction

Distributed data-parallel optimization algorithms help us tackle the increasing complexity of machine learning models and of the data on which they are trained. We can classify those training algorithms as either *centralized* or *decentralized*, and we often consider those settings to have different benefits over training 'alone'. In the *centralized* setting, workers compute gradients on independent mini-batches of data, and they average those gradients between all workers. The resulting lower variance in the updates enables larger learning rates and faster training. In the *decentralized* setting, workers average their models with only a sparse set of 'neighbors' in a graph instead of all-to-all, and they may have private datasets sampled from different distributions. As the benefit of decentralized learning, we usually focus only on the (indirect) access to other worker's datasets, and not of faster training.

While decentralized learning is typically studied with heterogeneous datasets across workers, sparse (decentralized) averaging between is also useful when worker's data is identically distributed (i.i.d.) [15]. As an example, sparse averaging is used in data centers to mitigate communication bottlenecks [1]. In fact the D-SGD algorithm [11], on which we focus in this work, performs well mainly in this setting, while algorithmic modifications [14, 21, 22] are required to yield good performance on heterogeneous objectives. When the data is i.i.d., the goal of sparse averaging is to optimize faster, just like in centralized (all-to-all) graphs.

Yet, current decentralized learning theory poorly explains the i.i.d. case. Analyses typically show that, for *small enough* learning rates, training with sparse averaging behaves the same as with all-to-all averaging [11, 8]. Compared to training alone with the *same small learning rate*, all-to-all averaging

[*] Equal contribution. Corresponding authors `thijs.vogels@epfl.ch` and `hadrien.hendrikx@epfl.ch`.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

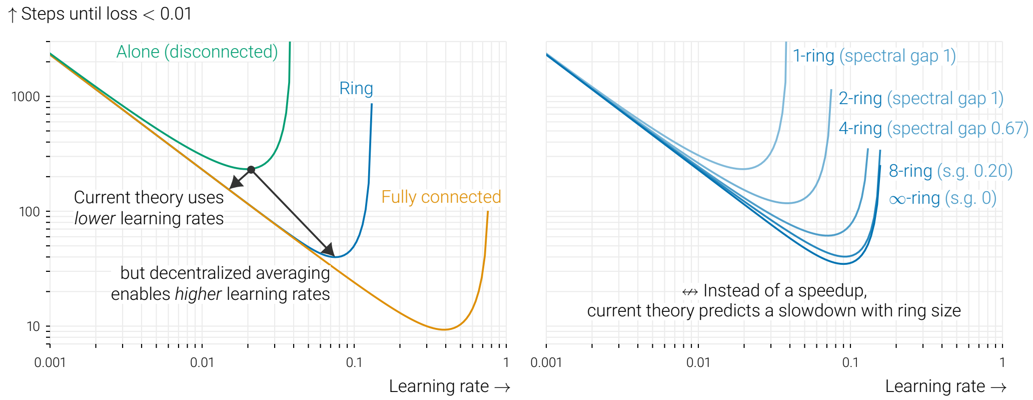

Figure 1: 'Time to target' for D-SGD [11] with constant learning rates on an i.i.d. isotropic quadratic dataset (Section 3). The noise disappears at the optimum. Compared to optimizing alone, 32 workers in a ring (*left*) are faster for any learning rate, but the largest improvement comes from being able to use a large learning rate. This benefit is not captured by current theory, which prescribes a smaller learning rate than training alone. On the *right*, we see that rings of increasing size enable larger learning rates and faster optimization. Because a ring's spectral gap goes to zero with the size, this cannot be explained by current theory.

reduces the gradient variance by the number of workers. In practice, however, such small learning rates would never be used. In fact, a reduction in variance should allow us to use a *larger* learning rate than training alone, rather than imposing a *smaller* one. Contrary to current theory, we show that averaging reduces the variance from the start, instead of just asymptotically. Lower variance increases the maximum learning rate, which directly speeds up convergence. We characterize how much averaging with various communication graphs reduces the variance, and show that centralized performance is not always achieved when using optimal large learning rates. The behavior we explain is illustrated in Figure 1.

In current convergence rates, the graph topology appears through the *spectral gap* of its averaging (gossip) matrix. The spectral gap poses a conservative lower bound on how much one averaging step brings all worker's models closer together. The larger, the better. If the spectral gap is small, a significantly smaller learning rate is required to make the algorithm behave close to SGD with all-to-all averaging with the same learning rate. Unfortunately, we experimentally observe that, both in deep learning and in convex optimization, the spectral gap of the communication graph is *not predictive* of its performance under realistically tuned learning rates.

The problem with the spectral gap quantity is clearly illustrated in a simple example. Let the communication graph be a ring of varying size. As the size of the ring increases to infinity, its spectral gap goes to zero, since it becomes harder and harder to achieve consensus between all the workers. This leads to the optimization progress predicted by current theory to go to zero as well. Yet, this behavior does not match the empirical behavior of the rings with i.i.d. data. As the size of the ring increases, the convergence rate actually *improves* (Figure 1), until it saturates at a point that depends on the problem.

In this work, we aim to accurately describe the behavior of i.i.d. distributed learning algorithms with sparse averaging, both in theory and in practice. We quantify the role of the graph in a quadratic toy problem designed to mimic the initial phase of deep learning (Section 3), showing that averaging enables a larger learning rate. From these insights, we derive a problem-independent notion of 'effective number of neighbors' in a graph that is consistent with time-varying topologies and infinite graphs, and is predictive of a graph's empirical performance in both convex and deep learning. We provide convergence proofs for convex and (strongly) convex objectives that only mildly depend on the spectral gap of the graph (Section 4), and consider the whole spectrum instead. At its core, our analysis does not enforce global consensus, but only between workers that are close to each other in the graph. Our theory shows that sparse averaging provably enables larger learning rates and thus speeds up optimization. These insights prove to be relevant in deep learning, where we accurately describe the performance of a variety of topologies, while their spectral gap does not (Section 5).

2 Related work

Decentralized SGD This paper studies decentralized SGD. Koloskova et al. [8] obtain the tightest bounds for this algorithm in the general setting where workers optimize heterogeneous objectives. Contrary to their work, we focus primarily on the case where all workers sample i.i.d. data from the same distribution. This important case is not described in a meaningful way by their analysis: while they show that gossip averaging reduces the asymptotic variance suffered by the algorithm, the fast initial linear decrease term in their convergence rate depends on the spectral gap of the gossip matrix. This key term does not improve through collaboration and gives rise to a *smaller learning rate* than training alone. Besides, as discussed above, this implies that optimization is not possible in the limit of large graphs, even in the absence of heterogeneity: for instance, the spectral gap of an infinite ring is zero, which would lead to a learning rate of zero as well.

These rates suggest that decentralized averaging speeds up the last part of training (dominated by variance), at the cost of slowing down the initial (linear convergence) phase. Beyond the work of Koloskova et al. [8], many papers focus on *linear speedup* (in the variance phase) over optimizing alone, and prove similar results in a variety of settings [11, 20, 12]. All these results rely on the following insight: while linear speedup is only achieved for small learning rates, SGD eventually requires such small learning rates anyway (because of, e.g., variance, or non-smoothness). This observation leads these works to argue that "topology does not matter". This is the case indeed, but only for very small learning rates, as shown in Figure 1. In practice, averaging speeds up both the initial *and* last part of training. This is what we show in this work, both in theory and in practice.

Another line of work studies D-(S)GD under statistical assumptions on the local data. In particular, Richards and Rebeschini [18] show favorable properties for D-SGD with graph-dependent implicit regularization and attain optimal statistical rates. Their suggested learning rate does depend on the spectral gap of the communication network, and it goes to zero when the spectral gap shrinks. Richards and Rebeschini [17] also show that larger (constant) learning rates can be used in decentralized GD, but their analysis focuses on decentralized kernel regression. It does not cover stochastic gradients, and relies on statistical concentration of local objectives rather than analysis on local neighborhoods.

Gossiping in infinite graphs An important feature of our results is that they only mildly depend on the spectral gap, and so they apply independently of the size of the graph. Berthier et al. [3] study acceleration of gossip averaging in infinite graphs, and obtain the same conclusions as we do: although spectral gap is useful for asymptotics, it fails to accurately describe the transient regime of averaging. This is especially limiting for optimization (compared to of just averaging), as new local updates need to be averaged at every step. The transient regime of averaging deeply matters. Indeed, it impacts the quality of the gradient updates, and so it rules the asymptotic regime of optimization.

The impact of the topology Some works on linear speedup [11] argue that the topology of the graph does not matter. This is only true for asymptotic rates in specific settings, as illustrated in Figure 1. Neglia et al. [16] investigate the impact of the topology on decentralized optimization, and contradict this claim. Compared to us, they make different noise assumptions, which in particular depend on the spectral distribution of the noise over the eigenvalues of the Laplacian (thus mixing computation and communication aspects). Although they show that the topology has an impact in the early phases of training (just like we do), they still get an unavoidable dependence on the spectral gap of the graph. Our results are different in nature, and show the benefits of averaging and the impact of the topology through the choice of large learning rates.

Another line of work studies the interaction of topology with particular patterns of data heterogeneity [4, 2], and how to optimize graphs with this heterogeneity in mind. These works "only" show a benefit from one-step gossip averaging and this is thus what they optimize the graph for. In contrast, we show that it is possible to benefit from distant workers beyond direct neighbors, too. This is an orthogonal direction, though the insights from our work could be used to strengthen their results.

Time-varying topologies Time-varying topologies are popular for decentralized deep learning in data centers due to their strong mixing [1, 23]. The benefit of varying the communication topology over time is not easily explained through standard theory, but requires dedicated analysis [25]. While our proofs only cover static topologies, the quantities that appear in our analysis can be computed for time-varying schemes, too. With these quantities, we can empirically study static and time-varying schemes in the same framework.

3 A toy problem: D-SGD on isotropic random quadratics

Before analyzing decentralized stochastic optimization through theory for general convex objectives and deep learning experiments, we first investigate a simple toy example that illustrates the behavior we want to explain in the analysis. In this setting, we can exactly characterize the convergence of decentralized SGD. We also introduce concepts that will be used throughout the paper.

We consider n workers that jointly optimize an isotropic quadratic $\mathbb{E}_{\mathbf{d}\sim\mathcal{N}^d(0,1)}\,\frac{1}{2}(\mathbf{d}^\top\mathbf{x})^2 = \frac{1}{2}\|\mathbf{x}\|^2$ with a unique global minimum $\mathbf{x}^\star = \mathbf{0}$. The workers access the quadratic through stochastic gradients of the form $\mathbf{g}(\mathbf{x}) = \mathbf{d}\mathbf{d}^\top\mathbf{x}$, with $\mathbf{d}\sim\mathcal{N}^d(0,1)$. This corresponds to a linear model with infinite data, and where the model can fit the data perfectly, so that stochastic noise goes to zero close to the optimum. We empirically find that this simple model is a meaningful proxy for the initial phase of (over-parameterized) deep learning (Section 5). A benefit of this model is that we can compute exact rates for it. These rates illustrate the behavior that we capture more generally in the theory of Section 4. Appendix C contains a detailed version of this section that includes full derivations.

The stochasticity in this toy problem can be quantified by the *noise level*

$$\zeta = \sup_{\mathbf{x}\in\mathbb{R}^d} \frac{\mathbb{E}_{\mathbf{d}}\|\mathbf{d}\mathbf{d}^\top\mathbf{x}\|^2}{\|\mathbf{x}\|^2}, \tag{1}$$

which is equal to $\zeta = d+2$, due to the random normal distribution of $\mathbf{d}$.

The workers run the D-SGD algorithm [11]. Each worker i has its own copy $\mathbf{x}_i \in \mathbb{R}^d$ of the model, and they alternate between local model updates $\mathbf{x}_i \leftarrow \mathbf{x}_i - \eta\mathbf{g}(\mathbf{x}_i)$ and averaging their models with others: $\mathbf{x}_i \leftarrow \sum_{j=1}^n w_{ij}\mathbf{x}_j$. The averaging weights w_{ij} are summarized in the *gossip matrix* $\mathbf{W}\in\mathbb{R}^{n\times n}$. A non-zero weight w_{ij} indicates that i and j are directly connected. In the following, we assume that $\mathbf{W}$ is symmetric and doubly stochastic: $\sum_{j=1}^n w_{ij} = 1\ \forall i$.

On our objective, D-SGD either converges or diverges linearly. Whenever it converges, i.e. when the learning rate is small enough, there is a convergence rate r such that

$$\mathbb{E}\|\mathbf{x}_i^{(t)}\|^2 \le (1-r)\|\mathbf{x}_i^{(t-1)}\|^2,$$

with equality as $t\to\infty$ (proofs in Appendix C). When the workers train alone ($\mathbf{W} = \mathbf{I}$), the convergence rate for a given learning rate η reads:

$$r_{\text{alone}} = 1 - (1-\eta)^2 - (\zeta-1)\eta^2. \tag{2}$$

The optimal learning rate $\eta^\star = \frac{1}{\zeta}$ balances the optimization term $(1-\eta)^2$ and the stochastic term $(\zeta-1)\eta^2$. In the centralized (fully connected) setting ($w_{ij} = \frac{1}{n}\ \forall i,j$), the rate is simple as well:

$$r_{\text{centralized}} = 1 - (1-\eta)^2 - \frac{(\zeta-1)\eta^2}{n}. \tag{3}$$

Averaging between n workers reduces the impact of the gradient noise, and the optimal learning rate grows to $\eta^\star = \frac{n}{n+\zeta-1}$. D-SGD with a general gossip matrix $\mathbf{W}$ interpolates those results.

To quantify the reduction of the $(\zeta-1)\eta^2$ term in general, we introduce the *problem-independent* notion of *effective number of neighbors* $n_{\mathbf{W}}(\gamma)$ of the gossip matrix $\mathbf{W}$ and *decay parameter* γ.

Definition A. The effective number of neighbors $n_{\mathbf{W}}(\gamma) = \lim_{t\to\infty}\frac{\sum_{i=1}^n \operatorname{Var}[\mathbf{y}_i^{(t)}]}{\sum_{i=1}^n \operatorname{Var}[\mathbf{z}_i^{(t)}]}$ measures the ratio of the asymptotic variance of the processes

$$\mathbf{y}^{(t+1)} = \sqrt{\gamma}\cdot\mathbf{y}^{(t)} + \boldsymbol{\xi}^{(t)}, \quad \text{where } \mathbf{y}^{(t)}\in\mathbb{R}^n \text{ and } \boldsymbol{\xi}^{(t)}\sim\mathcal{N}^n(0,1) \tag{4}$$

and

$$\mathbf{z}^{(t+1)} = \mathbf{W}(\sqrt{\gamma}\cdot\mathbf{z}^{(t)} + \boldsymbol{\xi}^{(t)}), \quad \text{where } \mathbf{z}^{(t)}\in\mathbb{R}^n \text{ and } \boldsymbol{\xi}^{(t)}\sim\mathcal{N}^n(0,1). \tag{5}$$

We call $\mathbf{y}$ and $\mathbf{z}$ *random walks* because workers repeatedly add noise to their state, somewhat like SGD's parameter updates. This should not be confused with a 'random walk' over nodes in the graph.

Since averaging with $\mathbf{W}$ decreases the variance of the random walk by at most n, the effective number of neighbors is a number between 1 and n. The decay γ modulates the sensitivity to communication delays. If $\gamma = 0$, workers only benefit from averaging with their direct neighbors. As γ increases,

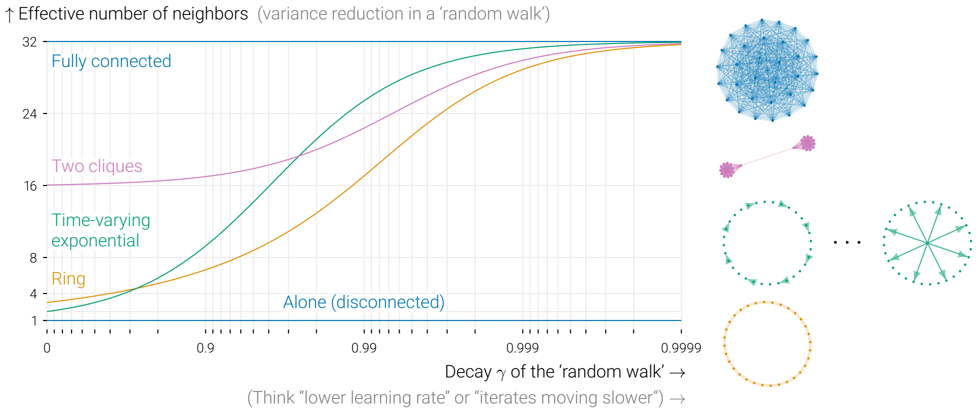

Figure 2: The effective number of neighbors for several topologies (Appendix B) measured by their variance reduction in Equation 5. The point γ on the x-axis that matters depends on the learning rate and the task. Which topology is 'best' varies from problem to problem. For large decay rates γ (corresponding small learning rates), all connected topologies achieve variance reduction close to a fully connected graph. For small decay rates (large learning rates), workers only benefit from their direct neighbors (e.g. 3 in a ring). These curves can be computed explicitly for constant topologies, and simulated efficiently for the time-varying exponential scheme [1].

multi-hop connections play an increasingly important role. As γ approaches 1, delayed and undelayed noise contributions become equally weighted, and the reduction tends to n for any connected topology.

For regular doubly-stochastic symmetric gossip matrices $\mathbf{W}$ with eigenvalues $\lambda_1, \ldots, \lambda_n$, $n_{\mathbf{W}}(\gamma)$ has a closed-form expression

$$n_{\mathbf{W}}(\gamma) = \frac{\frac{1}{1-\gamma}}{\frac{1}{n}\sum_{i=1}^{n} \frac{\lambda_i{}^2}{1-\lambda_i^2\gamma}}. \tag{6}$$

The notion of variance reduction in random walks, however, naturally extends to infinite topologies or time-varying averaging schemes as well. Figure 2 illustrates $n_{\mathbf{W}}$ for various topologies.

In our exact characterization of the convergence of D-SGD on the isotropic quadratic toy problem (Appendix C), we find that the effective number of neighbors appears in place of the number of workers n in the fully-connected rate of Equation 3. The rate is the unique solution to

$$r = 1 - (1-\eta)^2 - \frac{(\zeta-1)\eta^2}{n_{\mathbf{W}}\left(\frac{(1-\eta)^2}{1-r}\right)}. \tag{7}$$

For fully-connected and disconnected $\mathbf{W}$, $n_{\mathbf{W}}(\gamma) = n$ or 1 respectively, irrespective of γ, and Equation 7 recovers Equations 2 and 3. For other graphs, the effective number of workers depends on the learning rate. Current theory only considers the case where $n_{\mathbf{W}} \approx n$, but the small learning rates this requires can make the term $(1-\eta)^2$ too large, defeating the purpose of collaboration.

Beyond this toy problem, we find that the proposed notion of effective number of neighbors is also meaningful in the analysis of general objectives (Section 4) and in deep learning (Section 5).

4 Theoretical analysis

In the previous section, we have derived exact rates for a specific function. Now we present convergence rates for general (strongly) convex functions that are consistent with our observations in the previous section. We obtain rates that depend on the level of noise, the hardness of the objective, and the topology of the graph. We will assume the following randomized model for D-SGD:

$$\mathbf{x}_i^{(t+1)} = \begin{cases} \mathbf{x}_i^{(t)} - \eta \nabla f_{\xi_i^{(t)}}(\mathbf{x}_i^{(t)}) & \text{with probability } \frac{1}{2}, \\ \sum_{j=1}^{n} w_{ij} \mathbf{x}_j^{(t)} & \text{otherwise,} \end{cases} \tag{8}$$

where $f_{\xi_i^{(t)}}$ represent sampled data points and the gossip weights w_{ij} are elements of $\mathbf{W}$. This randomized model yields a clean analysis, but similar results hold for standard D-SGD (Appendix D.4).

Assumption B. The stochastic gradients are such that: (I) $\xi_i^{(t)}$ and $\xi_j^{(\ell)}$ are independent for all t, ℓ and $i \neq j$. (II) $\mathbb{E}\left[f_{\xi_i^{(t)}}\right] = f$ for all t, i (III) $\mathbb{E}\left\|\nabla f_{\xi_i^{(t)}}(\mathbf{x}^\star)\right\|^2 \leq \sigma^2$ for all t, i, where $\mathbf{x}^\star$ is a minimizer of f. (IV) $f_{\xi_i^{(t)}}$ is convex and ζ-smooth for all t, i. (V) f is μ-strongly-convex for $\mu \geq 0$ and L-smooth.

The smoothness ζ of the stochastic functions f_ξ defines the level of noise in the problem (the lower, the better). The ratio ζ/L compares the difficulty of optimizing with stochastic gradients to the difficulty with the true global gradient (before reaching the 'variance region' of distance $\mathcal{O}(\sigma^2)$ to the optimum). Assuming better smoothness for the global average objective than for the local functions is key to showing the benefit of averaging between workers. Without communication, convergence to the variance region is ensured for learning rates $\eta \leq 1/\zeta$. If $\zeta \approx L$, there is little noise and cooperation does not help before $\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2 \approx \sigma^2$. Yet, in noisy regimes ($\zeta \gg L$), such as in Section 3 in which $\zeta = d + 2 \gg 1 = L$, averaging enables larger step-sizes up to $\min(1/L, n/\zeta)$, greatly speeding up the initial training phase. This is precisely what we prove in Theorem I.

If the workers always remain close ($\mathbf{x}_i \approx \frac{1}{n}(\mathbf{x}_1 + \ldots + \mathbf{x}_n)$ $\forall i$, or equivalently $\frac{1}{n}\mathbf{1}\mathbf{1}^\top\mathbf{x} \approx \mathbf{x}$), D-SGD behaves the same as SGD on the average parameter $\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$, and the learning rate depends on $\max(\zeta/n, L)$, showing a reduction of variance by n. To maintain "$\frac{1}{n}\mathbf{1}\mathbf{1}^\top\mathbf{x} \approx \mathbf{x}$", however, we require a small learning rate. This is a common starting point for the analysis of D-SGD, in particular for the proofs in Koloskova et al. [8]. On the other extreme, if we do not assume closeness between workers, "$\mathbf{I}\mathbf{x} \approx \mathbf{x}$" always holds. In this case, there is no variance reduction, but no requirement for a small learning rate either. In Section 3, we found that, at the optimal learning rate, workers are *not* close to all other workers, but they *are* close to others that are not too far away in the graph.

We capture the concept of 'local closeness' by defining an averaging matrix $\mathbf{M}$. It allows us to consider semi-local averaging beyond direct neighbors, but without fully averaging with the whole graph. We ensure that "$\mathbf{M}\mathbf{x} \approx \mathbf{x}$", leading to some improvement in the smoothness between ζ and ζ/n, interpolating between the two previous cases. Each matrix $\mathbf{M}$ implies a requirement on the learning rate, as well as an improvement in smoothness. Based on Section 3, we therefore focus on a specific family of matrices that strike a good balance between the two: We choose $\mathbf{M}$ as the covariance of a decay-γ 'random walk process' with the graph, meaning that

$$\mathbf{M} = (1-\gamma)\sum_{k=1}^{\infty}\gamma^{k-1}\mathbf{W}^{2k} = (1-\gamma)\mathbf{W}^2(1-\gamma\mathbf{W}^2)^{-1}. \tag{9}$$

Varying γ induces a spectrum of averaging neighborhoods from $\mathbf{M} = \mathbf{W}^2$ ($\gamma = 0$) to $\mathbf{M} = \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ ($\gamma = 1$). γ also implies an effective number of neighbors $n_\mathbf{W}(\gamma)$: the larger γ, the larger $n_\mathbf{W}(\gamma)$.

Theorem I provides convergence rates for any value of γ, but the best rates are obtained for a specific γ that balances the benefit of averaging with the constraint it imposes on closeness between neighbors. In the following theorem, we assume that $\mathbf{M}_{ii} = \mathbf{M}_{jj}$ for all i, j, so that ${\mathbf{M}_{ii}}^{-1} = n_\mathbf{W}(\gamma)$: the effective number of neighbors defined in (6) is equal to the inverse of the self-weights of matrix $\mathbf{M}$. Otherwise, all results hold by replacing $n_\mathbf{W}(\gamma)$ with $\min_i {\mathbf{M}_{ii}}^{-1}$.

Theorem I. If Assumption B holds, and the learning rate satisfies

$$\eta \leq \min\left(\frac{1}{8(\zeta/n_\mathbf{W}(\gamma) + L)}, \frac{1-\gamma\lambda_2(\mathbf{W})}{2n_\mathbf{W}(\gamma)L}\right), \tag{10}$$

then the iterates obtained by (8) verify

$$\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|_\mathbf{M}^2 + \frac{1}{n_\mathbf{W}(\gamma)}\|\mathbf{x}^{(t)}\|_{\mathbf{I}-\mathbf{M}}^2 \leq \left(1 - \frac{\eta\mu}{2}\right)^t C_0 + \frac{8\eta\sigma^2}{n_\mathbf{W}(\gamma)}, \tag{11}$$

The bound on the learning rate (10) represents the tension between (I) reducing the noise ζ by averaging with more people (larger $n_\mathbf{W}(\gamma)$), which is the first term in the minimum, and (II) staying close to all of them. A large spectral gap $1 - \lambda_2(\mathbf{W})$ reduces the second constraint, but we allow non-trivial learning rates $\eta > 0$ even when $\lambda_2(\mathbf{W}) = 1$ (infinite graphs) as long as $\gamma < 1$.

Theorem I gives a rate for each parameter γ that controls the local neighborhood size. The task that remains is to find the γ parameter that gives the best convergence guarantees (the largest learning rate). As explained before, one should never reduce the learning rate in order to be close to others,

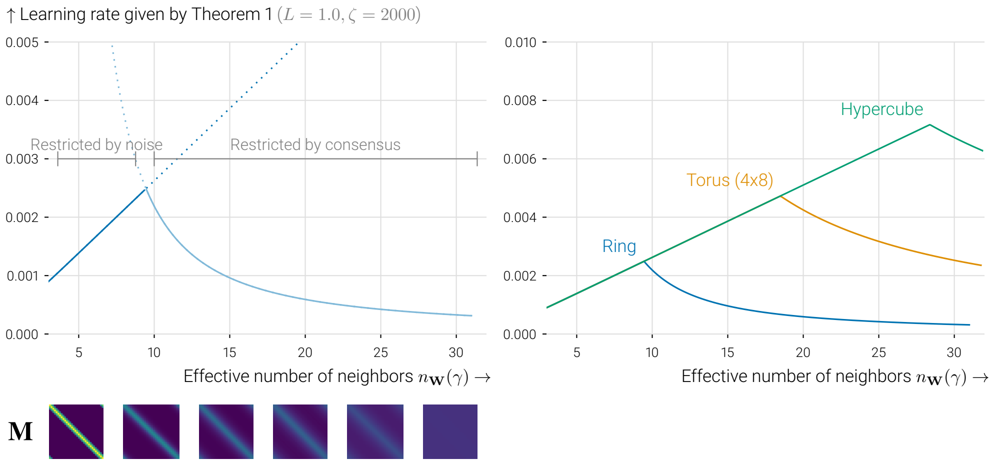

Figure 3: Maximum learning rates prescribed by Theorem I, varying the parameter γ that implies an effective neighborhood size (x-axis) and an averaging matrix $\mathbf{M}$ (drawn as *heatmaps*). On the *left*, we show the details for a 32-worker ring topology, and on the *right*, we compare it to more connected topologies. Increasing γ (and with it $n_{\mathbf{W}}(\gamma)$) initially leads to larger learning rates thanks to noise reduction. At the optimum, the cost of consensus exceeds the benefit of further reduced noise.

because the goal of collaboration is to *increase* the learning rate. We should therefore pick γ such that the first term in Equation (10) dominates. This intuition is summarized in Corollary II, which compares the performance of D-SGD with centralized SGD with fewer workers.

Corollary II. D-SGD is as fast as centralized mini-batch SGD with $\mathcal{O}(n_{\mathbf{W}}(\gamma))$ workers, assuming that $\zeta \geq nL$, and that the parameter γ is the highest γ such that $\frac{2n_{\mathbf{W}}(\gamma)^2}{1-\gamma\lambda_2(\mathbf{W})} \leq 32\frac{\zeta}{L}$. This corresponds to a learning rate $\eta = n_{\mathbf{W}}(\gamma)/16\zeta$.

The typical D-SGD learning rates [8] are of order $\mathcal{O}(\min(1/T, 1-\lambda_2(W)))$, which are much smaller than the learning rate of Corollary II when $\lambda_2(\mathbf{W})$ is large or the number of iterations large. We use the condition $\zeta \geq nL$ only to present results in a simpler way. The condition $\frac{2n_{\mathbf{W}}(\gamma)^2}{1-\gamma\lambda_2(\mathbf{W})}$ only depends on the size and topology of the graph, and can easily be computed in many cases. Thus, to obtain the best guarantees, we start from $\gamma = 0$ and then increase it until either $n_{\mathbf{W}}(\gamma) \approx n$, the total size of the graph, or the two terms in the minimum match. This is how we obtain Figure 3.

Proof sketch (Theorem I). The proof relies on a simple argument: rather than bounding $\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2$ or $\|\frac{1}{n}\mathbf{1}\mathbf{1}^\top\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2$, we analyze $\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|_{\mathbf{M}}^2$. This term better captures the benefit of averaging than $\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2$, thus leading to better smoothness constants, as long as $\|\mathbf{x}^{(t)}\|_{\mathbf{I}-\mathbf{M}}^2$ is not too large. This yields fast rates without the need to guarantee that iterates between very distant workers remain close, which would be prohibitively expensive. □

Theorem I is a special case of a more general theorem presented in Appendix D. This version, among other things, covers different choices of parameters, unbalanced communication and computation probabilities (thus allowing for local steps), and the convex ($\mu = 0$) case.

5 Experimental analysis

While in the previous sections we have discussed isotropic quadratics or convex and smooth functions, the initial motivation for this work comes from observations in deep learning. First, it is crucial in deep learning to use a large learning rate in the initial phase of training [10]. Contrary to what current theory prescribes, we do not use smaller learning rates in decentralized optimization than when training alone (even when data is heterogeneous.) And second, we find that the spectral gap of a topology is not predictive of the performance of that topology in deep learning experiments.

In this section, we experiment with a variety of 32-worker topologies on Cifar-10 [9] with a VGG-11 model [19]. Like other recent works [13, 22], we opt for this older model, because it does not

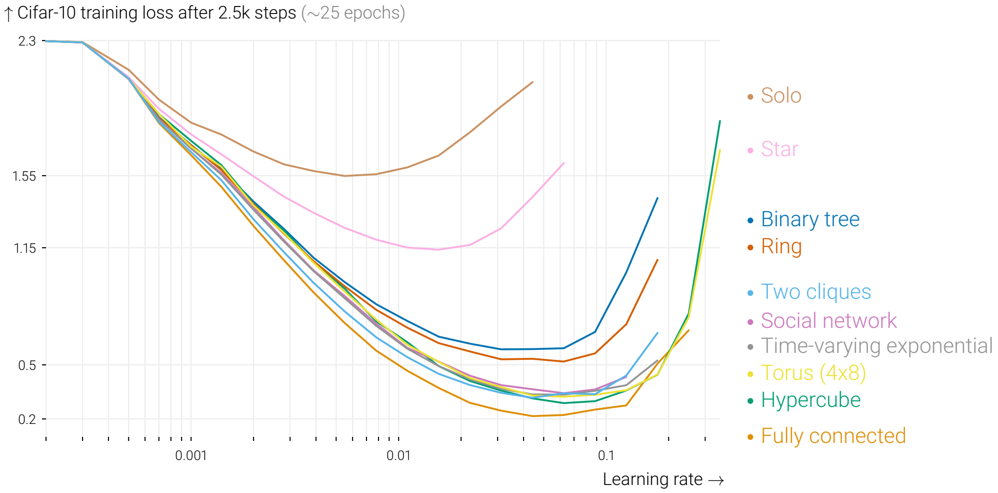

Figure 4: Training loss reached after 2.5 k SGD steps with a variety of graph topologies. In all cases, averaging yields a small increase in speed for small learning rates, but a large gain over training alone comes from being able to increase the learning rate. While the star has a better spectral gap (0.031) than the ring (0.013), it performs worse, and does not allow large learning rates. For reference, similar curves for fully-connected graphs of varying sizes are in Appendix F.

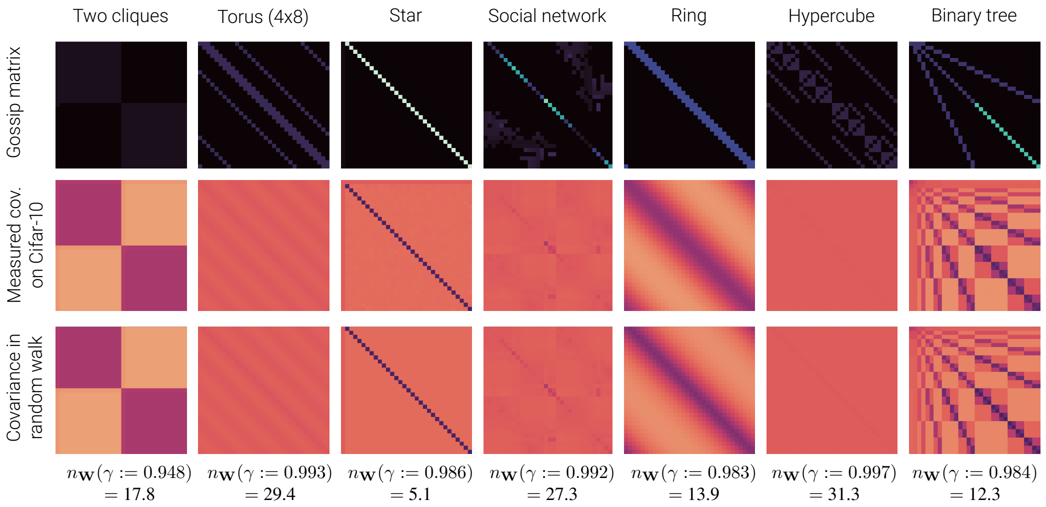

Figure 5: Measured covariance in Cifar-10 (second row) between workers using various graphs (top row). After 10 epochs, we store a checkpoint of the model and train repeatedly for 100 SGD steps, yielding 100 models for 32 workers. We show normalized covariance matrices between the workers. These are very well approximated by the covariance in the random walk process of Section 3 (third row). We print the fitted decay parameters and corresponding 'effective number of neighbors'.

include BatchNorm [6] which forms an orthogonal challenge for decentralized SGD. Please refer to Appendix E for full details on the experimental setup. Our set of topologies (Appendix B) includes regular graphs like rings and toruses, but also irregular graphs such as a binary tree [22] and social network [5], and a time-varying exponential scheme [1]. We focus on the initial phase of training, 25k steps in our case, where both train and test loss converge close to linearly. Using a large learning rate in this phase is found to be important for good generalization [10].

Figure 4 shows the loss reached after the first 2.5 k SGD steps for all topologies and for a dense grid of learning rates. The curves have the same global structure as those for isotropic quadratics Figure 1: (sparse) averaging yields a small increase in speed for small learning rates, but a large gain over training alone comes from being able to increase the learning rate. The best schemes support almost the same learning rate as 32 fully-connected workers, and get close in performance.

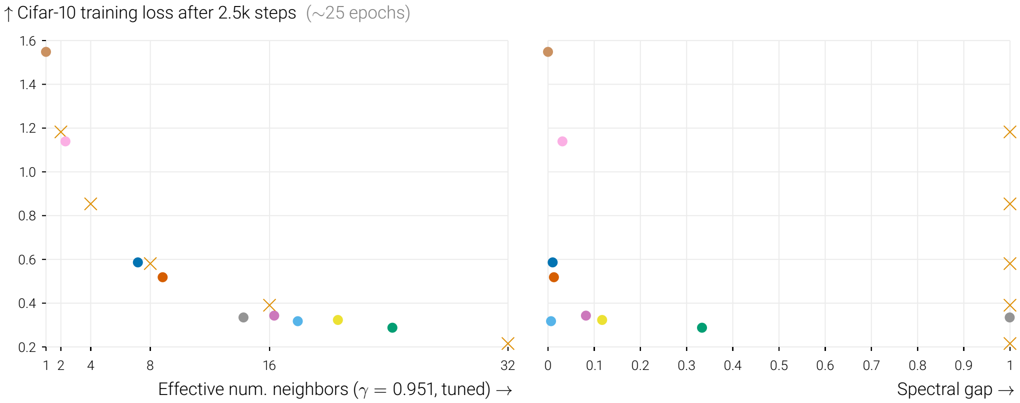

Figure 6: Cifar-10 training loss after 2.5 k steps for all studied topologies with their optimal learning rates. Colors match Figure 4, and × indicates fully-connected graphs with varying number of workers. After fitting a decay parameter $\gamma = 0.951$ that captures problem specifics, the effective number of neighbors (left) as measured by variance reduction in a random walk (like in Section 3) explains the relative performance of these graphs much better than the spectral gap of these topologies (right).

We also find that the random walks introduced in Section 3 are a good model for variance between workers in deep learning. Figure 5 shows the empirical covariance between the workers after 100 SGD steps. Just like for isotropic quadratics, the covariance is accurately modeled by the covariance in the random walk process for a certain decay rate γ.

Finally, we observe that the effective number of neighbors computed by the variance reduction in a random walk (Section 3) accurately describes the relative performance under tuned learning rates of graph topologies on our task, including for irregular and time-varying topologies. This is in contrast to the topology's spectral gaps, which we find to be not predictive. We fit a decay rate $\gamma = 0.951$ that seems to capture the specifics of our problem, and show the correlation in Figure 6.

In Appendix F.1, we replicate the same experiments in a different setting. There, we use larger graphs (of 64 workers), a different model and data set (an MLP on Fashion MNIST [24]), and no momentum or weight decay. The results in this setting are qualitatively comparable to the ones presented above.

6 Conclusion

We have shown that the sparse averaging in decentralized learning allows larger learning rates to be used, and that it speeds up training. With the optimal large learning rate, the workers' models are not guaranteed to remain close to their global average. Enforcing global consensus is unnecessary in the i.i.d. setting and the small learning rates it would require are counter-productive. With the optimal learning rate, models *do* remain close to some local average in a weighted neighborhood around them. The workers benefit from a number of 'effective neighbors', smaller than the whole graph, that allow them to use a large learning rate while retaining sufficient consensus within the 'local neighborhood'.

Based on our insights, we encourage practitioners of sparse distributed learning to look beyond the spectral gap of graph topologies, and to investigate the actual 'effective number of neighbors' that is used. We also hope that our insights motivate theoreticians to be mindful of assumptions that artificially limit the learning rate.

We show experimentally that our conclusions hold in deep learning, but extending our theory to the non-convex setting is an important open direction that could reveal interesting new phenomena. Furthermore, an extension of our semi-local analysis to the heterogeneous setting where workers optimize different objectives could shed further light on the practical performance of D-SGD.

Acknowledgments and Disclosure of Funding

This project was supported by SNSF grant 200020_200342.

We thank Lie He for valuable conversations and for identifying the discrepancy between a topology's spectral gap and its empirical performance. We also thank Raphaël Berthier, Aditya Vardhan Varre and Yatin Dandi for their feedback on the manuscript.

References

[1] Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Michael G. Rabbat. Stochastic gradient push for distributed deep learning. In *Proc. ICML*, volume 97, pages 344–353, 2019.

[2] B. Le Bars, Aurélien Bellet, Marc Tommasi, and Anne-Marie Kermarrec. Yes, topology matters in decentralized optimization: Refined convergence and topology learning under heterogeneous data. *CoRR*, abs/2204.04452, 2022.

[3] Raphaël Berthier, Francis R. Bach, and Pierre Gaillard. Accelerated gossip in networks of given dimension using jacobi polynomial iterations. *SIAM J. Math. Data Sci.*, 2(1):24–47, 2020.

[4] Yatin Dandi, Anastasia Koloskova, Martin Jaggi, and Sebastian U. Stich. Data-heterogeneity-aware mixing for decentralized learning. *CoRR*, abs/2204.06477, 2022.

[5] Allison Davis, Burleigh Bradford Gardner, and Mary R Gardner. *Deep South: A social anthropological study of caste and class*. Univ of South Carolina Press, 1930.

[6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, volume 37, pages 448–456, 2015.

[7] Sham Kakade, Shai Shalev-Shwartz, Ambuj Tewari, et al. On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization. *Unpublished Manuscript*, 2(1):35, 2009.

[8] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U. Stich. A unified theory of decentralized SGD with changing topology and local updates. In *Proc. ICML*, volume 119, pages 5381–5393, 2020.

[9] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (Canadian Institute for Advanced Research).

[10] Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. In *NeurIPS*, pages 11669–11680, 2019.

[11] 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 *NeurIPS*, pages 5330–5340, 2017.

[12] Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. In *Proc. ICML*, volume 80, pages 3049–3058, 2018.

[13] Tao Lin, Sai Praneeth Karimireddy, Sebastian U. Stich, and Martin Jaggi. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. In *Proc. ICML*, volume 139, pages 6654–6665, 2021.

[14] Paolo Di Lorenzo and Gesualdo Scutari. NEXT: in-network nonconvex optimization. *IEEE Trans. Signal Inf. Process. over Networks*, 2(2):120–136, 2016.

[15] Yucheng Lu and Christopher De Sa. Optimal complexity in decentralized training. In *Proc. ICML*, volume 139, pages 7111–7123, 2021.

[16] Giovanni Neglia, Chuan Xu, Don Towsley, and Gianmarco Calbi. Decentralized gradient methods: does topology matter? In *AISTATS,*, volume 108, pages 2348–2358, 2020.

[17] Dominic Richards and Patrick Rebeschini. Optimal statistical rates for decentralised non-parametric regression with linear speed-up. In *NeurIPS*, pages 1214–1225, 2019.

[18] Dominic Richards and Patrick Rebeschini. Graph-dependent implicit regularisation for distributed stochastic subgradient descent. *J. Mach. Learn. Res.*, 21:34:1–34:44, 2020.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[20] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. d^2: Decentralized training over decentralized data. In *Proc. ICML*, volume 80, pages 4855–4863, 2018.

[21] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D^2: Decentralized training over decentralized data. In *Proc. ICML*, volume 80, pages 4855–4863, 2018.

[22] Thijs Vogels, Lie He, Anastasia Koloskova, Sai Praneeth Karimireddy, Tao Lin, Sebastian U. Stich, and Martin Jaggi. Relaysum for decentralized deep learning on heterogeneous data. In *NeurIPS*, pages 28004–28015, 2021.

[23] Jianyu Wang, Anit Kumar Sahu, Zhouyi Yang, Gauri Joshi, and Soummya Kar. MATCHA: speeding up decentralized SGD via matching decomposition sampling. *CoRR*, abs/1905.09435, 2019.

[24] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

[25] Bicheng Ying, Kun Yuan, Yiming Chen, Hanbin Hu, Pan Pan, and Wotao Yin. Exponential graph is provably efficient for decentralized deep training. In *NeurIPS*, pages 13975–13987, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Contents of the Appendix

A Notation

Table 1 defines some notation and conventions used throughout this paper and in the appendix.

Table 1: Notation

Bold symbol $\mathbf{v}$	Vector
Bold uppercase $\mathbf{M}$	Matrix
$\mathcal{N}^d(0,1)$	Standard normal distribution with d independent dimensions
$\langle \mathbf{x}, \mathbf{y} \rangle$	Inner product $\mathbf{x}^\top \mathbf{y}$
$\lVert \mathbf{T} \rVert_2$	Spectral norm
$\lVert \mathbf{T} \rVert_F$	Frobenius norm
$\mathbf{P} \otimes \mathbf{Q}$	Kronecker product
$\mathbf{1}$	Vector of all ones

B Topologies

The static topologies that we consider in this work are drawn in Figure 7. Figures 8 and 9 show the gossip matrices we use in detail.

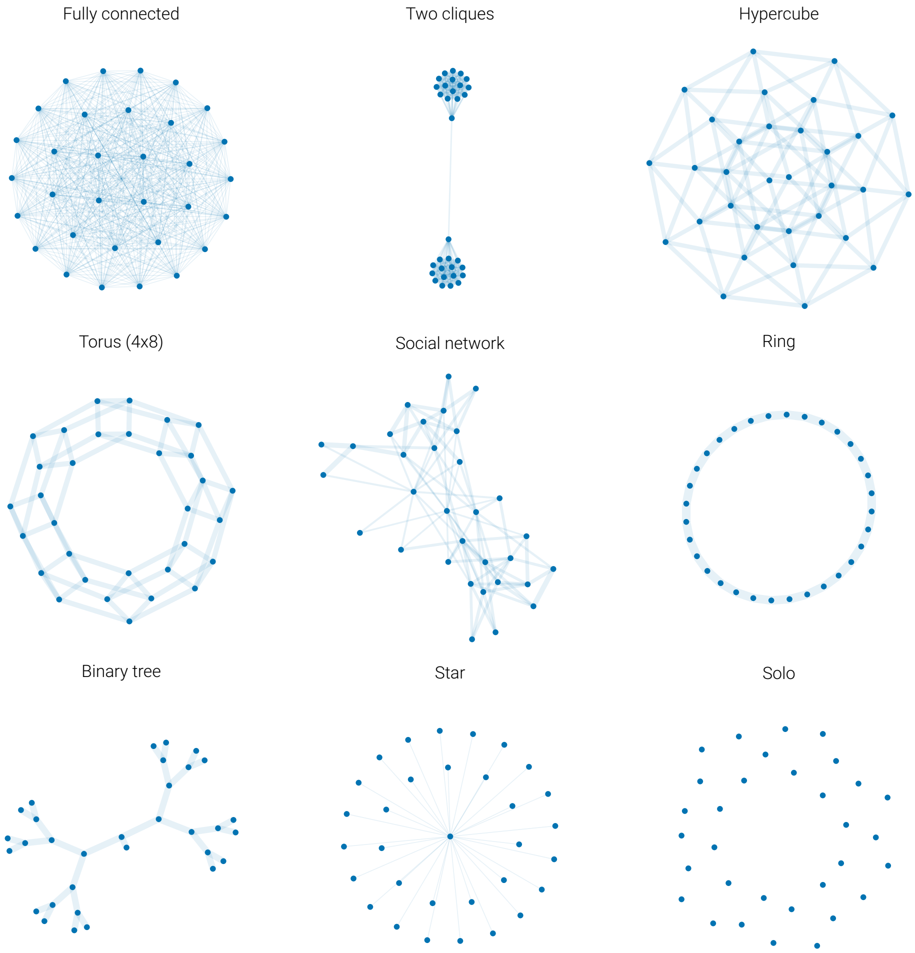

Figure 7: Spring-layout drawings of the static graph topologies considered used this paper. The nodes represent workers, and an edge between two workers indicates that they are connected. The thickness of a edges is proportional to its averaging weight.

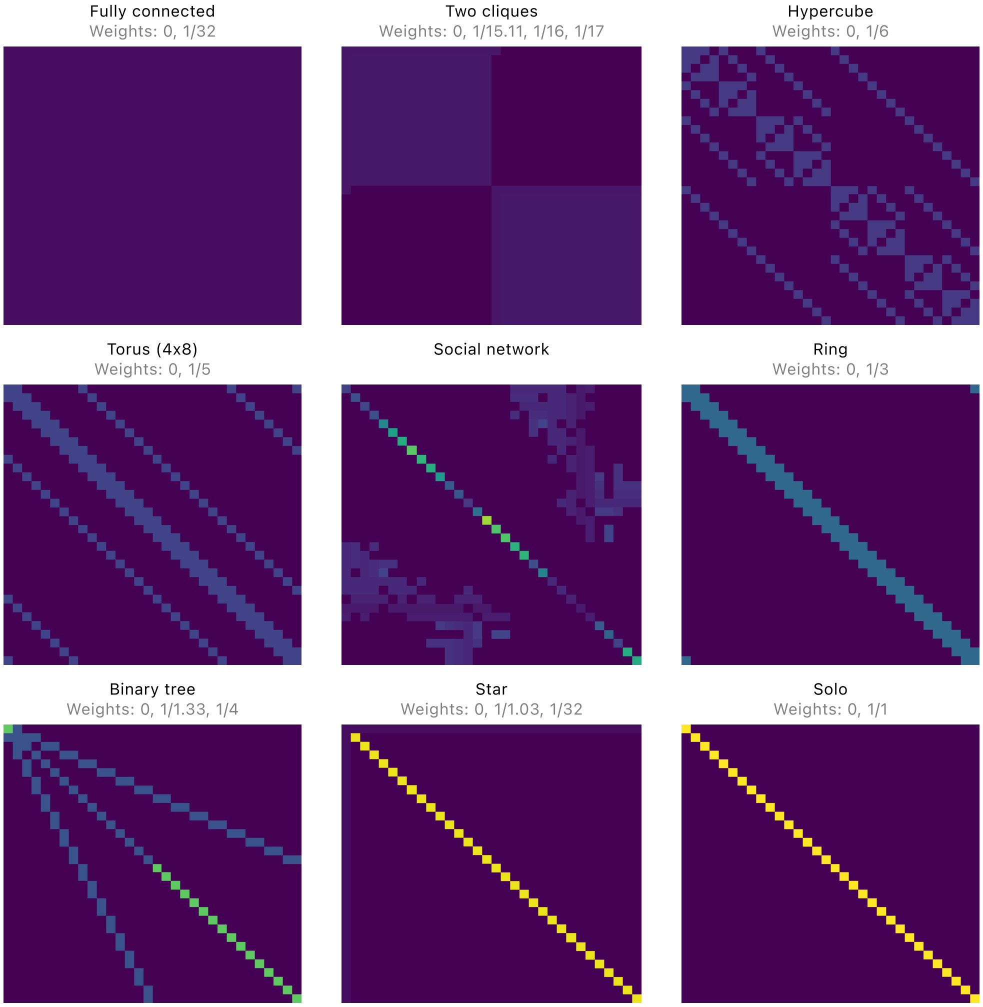

Figure 8: Gossip matrices corresponding to the graph topologies drawn in Figure 7. x and y axes represent workers, and the color of each coordinate in the plots indicates the gossip weight between each pair of workers. The brigher, the higher the weight.

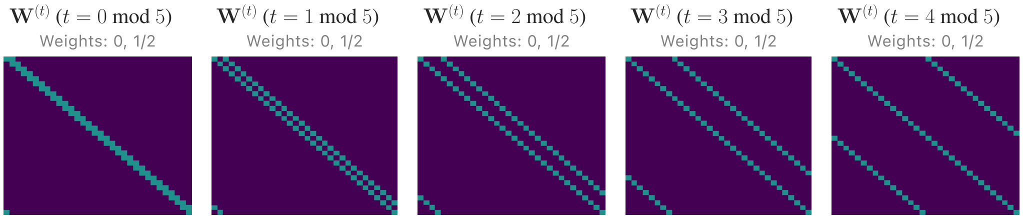

Figure 9: Gossip matrices for the time-varying exponential graph [1, 25]. The product of $\log n$ consecutive gossip matrices equals to the fully-connected averaging matrix with $w_{ij} = 1/n\ \forall i, j$.

C Random quadratics

C.1 Objective

We study the simple problem of minimizing an isotropic d-dimensional quadratic,

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$$

where the objective function $f(x) = \frac{1}{2}\|\mathbf{x}\|^2$ is considered to be the expectation over an infinite dataset with random normal features and labels 0:

$$f(\mathbf{x}) = \mathbb{E}_{\mathbf{d}\sim\mathcal{N}^d(0,1)} \frac{1}{2} \langle \mathbf{d}, \mathbf{x}\rangle^2 . \tag{12}$$

The optimum of this objective is at $\mathbf{x}^\star = \mathbf{0}$ without loss of generality, because any shifted quadratic would behave the same in the algorithm studied. We will access this objective function through stochastic gradients of the form $\mathbf{g}(\mathbf{x}) = \mathbf{d}\mathbf{d}^\top\mathbf{x}$. The stochasticity of these gradients disappears at the optimum, like in an over-parameterized model.

The difficulty of this problem depends on the dimensionality d. For a lower-dimensional problem, the 'stochastic Hessian' $\mathbf{d}\mathbf{d}^\top$ is closer to the true hessian $\mathbf{I}$ than for a high dimensional one. This level of stochasticity is captured by the following quantity:

Definition C (Noise level). $\zeta = \sup_{\mathbf{x}} \frac{\mathbb{E}_{\mathbf{d}}\|\mathbf{d}\mathbf{d}^\top\mathbf{x}\|^2}{\|\mathbf{x}\|^2}$.

For our random normal data with batch size 1, this notion of noise level corresponds directly to the dimensionality of the data as $\zeta = d + 2$.

C.2 Algorithm

The objective (12) is collaboratively optimized by n workers. At every time step t, each worker i has its own copy of the 'model' $\mathbf{x}_i^{(t)} \in \mathbb{R}^d$. In the D-SGD algorithm, workers iteratively compute stochastic gradient estimates $\mathbf{g}_i^{(t)} = \mathbf{d}_i^{(t)}\mathbf{d}_i^{(t)\top}\mathbf{x}_i^{(t)}$, where $\mathbf{d}_i^{(t)}$ are i.i.d. from $\mathcal{N}^d(0,1)$. The stochastic gradients are unbiased: $\mathbb{E}\,\mathbf{g}_i^{(t)} = \nabla f(\mathbf{x}_i^{(t)}) = \mathbf{x}_i^{(t)}$.

Workers interleave stochastic gradient updates with gossip averaging:

$$\mathbf{x}_i^{(0)} = \mathbf{x}^{(0)} \quad \forall i$$
$$\mathbf{x}_i^{(t+1)} = \mathbf{W}(\mathbf{x}_i^{(t)} - \eta\mathbf{g}_i^{(t)}),$$

where η is the learning rate and

$$\mathbf{W}(\mathbf{x}_i) = \sum_{j=1}^n w_{ij}\mathbf{x}_j .$$

This linear operation can be interpreted as matrix multiplication, but one operating on each coordinate of the model independently. $\mathbf{W}$ is an $n\times n$ matrix, and *not* a $d\times d$ matrix as the notation may suggest. The averaging weights w_{ij} encode the connectivity of the communication topology: non-zero w_{ij} implies that workers i and j are directly connected. We make several assumptions about the gossip weights in this analysis:

Assumption D. Constant gossip weights: The weights w_{ij} do not change between steps of D-SGD.

Assumption E. Symmetric gossip weights: $w_{ij} = w_{ji}$.

Assumption F. Doubly stochastic gossip weights: $w_{ij} \geq 0\ \forall i,j$, $\sum_j w_{ij} = 1\ \forall i$, $\sum_i w_{ij} = 1\ \forall j$.

Assumption G. Regular topology: all workers have k directly-connected neighbors, and $w_{ij} = c$ for some constant c, and for each edge where $i \neq j$.

Definition H. Spectrum of $\mathbf{W}$. Let the eigenvalues of $\mathbf{W}$ be $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. We call the corresponding eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$. Under assumption F, $\lambda_1 = 1$, and we call $1 - \lambda_2$ the *spectral gap* of $\mathbf{W}$.

The assumptions on constant gossip weights and regular topologies are mainly here to ease the analysis. We experimentally observe that our findings hold for time-varying topologies and infinite graphs, too, and that they approximately hold for irregular graphs.

C.3 Linear convergence of an unrolled error vector

We will study the convergence of the algorithm by tracking the *error matrix* $\mathbf{E} \in \mathbb{R}^{n \times n}$. The coordinates of this matrix are the expected covariance between each pair of workers in the network.

$$\mathbf{E}_{ij}^{(t)} = \mathbb{E}\left\langle \mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)} \right\rangle.$$

We sometimes flatten the error matrix into a vector $\mathbf{e} \in \mathbb{R}^{n^2}$, such that $\mathbf{e}_{ni+j} = \mathbf{E}_{ij}$. The diagonal entries of this matrix describe the worker's error on the objective, and as all workers converge to the optimum at zero, each entry of the matrix will converge to zero. Our analysis of $\mathbf{E}$ quantity starts with a key observation:

Lemma 1. There exists an $n^2 \times n^2$ 'transition' matrix $\mathbf{T}$ such that $\mathbf{e}^{(t+1)} = \mathbf{T}\mathbf{e}^{(t)}\ \forall t$.

Proof. Because both gossip averaging and the gradient updates are linear, this follows from expanding the inner product. □

The transition matrix $\mathbf{T}$ depends on the gossip matrix $\mathbf{W}$ and on the learning rate η. Its spectral gap describes the convergence of the algorithm. D-SGD converges linearly if the norm $\|\mathbf{T}\|_2 < 1$.

We separate $\mathbf{T}$ into a product $\mathbf{T} = \mathbf{T}^{\text{gossip}}\mathbf{T}^{\text{grad}}$, where $\mathbf{T}^{\text{grad}}$ and $\mathbf{T}^{\text{gossip}}$ respectively capture the gradient update and gossip steps of the algorithm. We find that

$$\mathbf{T}^{\text{gossip}} = \mathbf{W} \otimes \mathbf{W}$$

and that $\mathbf{T}^{\text{grad}}$ is diagonal. It only operates element-wise, such that

$$\left[\mathbf{T}^{\text{grad}}\mathbf{e}\right]_{ni+j} = \begin{cases} (1-\eta)^2\mathbf{e}_{ni+j} + (\zeta - 1)\eta^2\mathbf{e}_{ni+j} & i = j \text{ (same worker)}, \\ (1-\eta)^2\mathbf{e}_{ni+j} & i \neq j \text{ (different workers)}. \end{cases} \tag{13}$$

This follows directly from expanding the inner product $\langle \mathbf{x}_i - \eta\mathbf{g}_i, \mathbf{x}_j - \eta\mathbf{g}_j \rangle$. The terms with $i = j$ behave differently than the ones where $i \neq j$, because the noise cancels if $i \neq j$.

C.4 Random walks with gossip averaging

Before we study the convergence of D-SGD on the random quadratic objective, we first take a step back and inspect a particular random walk process, where workers average their random walk iterates through gossip averaging.

Let $\mathbf{z}^{(t)} \in \mathbb{R}^n$ be a vector containing (scalar) iterates of n workers in the following process:

$$\mathbf{z}^{(0)} = \mathbf{0}, \tag{14}$$

$$\mathbf{z}^{(t+1)} = \mathbf{W}\left(\sqrt{\gamma}\mathbf{z}^{(t)} + \boldsymbol{\xi}^{(t)}\right) \quad \text{where } \boldsymbol{\xi}^{(t)} \sim \mathcal{N}^n(0, 1). \tag{15}$$

We call the parameter $0 < \gamma \leq 1$ the 'decay rate'. Note that the name *random walk* refers to iterative addition of random noise to the workers iterates, and not to a 'random walk' between nodes of the graph.

For this random walk, we will track the covariance matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ across workers (and its flattened version $\mathbf{c} \in \mathbb{R}^{n^2}$). Its coordinates are

$$\mathbf{C}_{ij}^{(t)} = \mathbb{E}[\mathbf{z}_i^{(t)}\mathbf{z}_j^{(t)}].$$

Lemma 2. For static, symmetric and doubly-stochastic topologies (Assumptions D, E and F), the Eigen decomposition of the covariance is

$$\mathbf{C}^{(t)} = \sum_{i=1}^{n} c_i^{(t)}\mathbf{v}_i\mathbf{v}_i^\top,$$

with $0 \leq c_i^{(t)} \leq \frac{\lambda_i^2}{1-\gamma\lambda_i^2}$. Here $(\lambda_i, \mathbf{v}_i)$ are the eigenvalue/eigenvector pairs of $\mathbf{W}$. As $t \to \infty$, $c_i^{(t)} = \frac{\lambda_i^2}{1-\gamma\lambda_i^2}$ with equality.

Proof. We can unroll the iterations:

$$\mathbf{z}^{(t)} = \sum_{k=1}^{t} \mathbf{W}^k \gamma^{(k-1)/2} \boldsymbol{\xi}^{(t-k)}$$

and use the temporal independence of $\boldsymbol{\xi}^{(t)}$ to write

$$\mathbf{C}^{(t)} = \mathbb{E}[\mathbf{z}^{(t)} {\mathbf{z}^{(t)}}^\top] = \sum_{k=1}^{t} \gamma^k \mathbf{W}^{2k}\, \mathbb{E}[\boldsymbol{\xi}^{(t-k)} {\boldsymbol{\xi}^{(t-k)}}^\top] = \sum_{k=1}^{t} \gamma^k \mathbf{W}^{2k}.$$

Using commutativity of $\mathbf{W}$ and its Eigen decomposition (Assumptions E, F), we can decompose it as

$$\mathbf{C}^{(t)} = \sum_{i=1}^{n} \mathbf{v}_i \mathbf{v}_i^\top \underbrace{\left(\sum_{k=1}^{t} \gamma^{k-1} \lambda_i^{2k} \right)}_{c_i^{(t)}}$$

Because all terms of parenthesized expression are non-negative, and its limit equals $\frac{\lambda_i^2}{1-\gamma\lambda_i^2}$, this proves the Lemma. □

Lemma 3. When the topology is regular (Assumption G) in addition to the assumptions of Lemma 2, workers in the random walk process have equal variance:

$$\operatorname{Var}[\mathbf{z}_i^{(t)}] = \frac{1}{n} \operatorname{Tr}[\mathbf{C}^{(t)}] = \frac{1}{n} \sum_{i=1}^{n} c_i^{(t)}.$$

Proof. The variances of $\mathbf{z}_i$ are the diagonal entries of the covariance matrix. By regularity, and because workers are initialized equally, all workers should have the same variance. $\operatorname{Var}[\mathbf{z}_i^{(t)}]$ is therefore equal to the average diagonal entry of $\mathbf{C}_i^{(t)}$. The second equality is a standard property of the trace. □

Lemma 4. Under the assumptions of Lemma 3, the variance $\operatorname{Var}[\mathbf{z}_i^{(t)}]$ increases over time:

$$\operatorname{Var}[\mathbf{z}_i^{(t)}] \leq \operatorname{Var}[\mathbf{z}_i^{(t+1)}] \leq \lim_{t' \to \infty} \operatorname{Var}[\mathbf{z}_i^{(t')}] \qquad \forall t.$$

Proof. If we write $\operatorname{Var}[\mathbf{z}_i^{(t)}]$ as $\frac{1}{n}\sum_{i=1}^{n} c_i^{(t)}$ using Lemma 3, the statement of this Lemma follows from the realization in Lemma 2 that $c_i^{(t)}$ increases over time to the limit $\frac{\lambda_i^2}{1-\gamma\lambda_i^2}$ for all i. □

Note that while the results above are for static gossip matrices, random walks and these variance quantities can be analogously defined time-varying topologies. Those just lack a simple exact form. The stronger the averaging of the gossip process, the lower the variance. We capture this in the following quantity:

Definition I (Effective number of neighbors)**.**

$$n_{\mathbf{W}}(\gamma) = \frac{\frac{1}{1-\gamma}}{\lim_{t\to\infty} \frac{1}{n}\sum_{i=1}^{n} \operatorname{Var}[\mathbf{z}_i^{(t)}]},$$

where $\mathbf{z}$ are the iterates from a random walk with gossip averaging, with decay parameter γ. The numerator is the variance of the random walk process without any gossip averaging ($\mathbf{W} = \mathbf{I}$).

C.5 Converging random walk

The covariance of the random walk process $\mathbf{C}$ and the error matrix of D-SGD iterates $\mathbf{E}$ share clear similarities. The quantities are both iteratively updated by an affine transformation. The main difference between them, however, is that $\mathbf{C}$ converges to a non-zero constant while $\mathbf{E}$ converges linearly to zero (or it diverges.)

In the next section, we draw a clear connection between the two processes, but first, we define a modified version of the random walk process that further highlights their similarity.

Definition J (Scaled random walk)**.** Let $0 < r < 1$ be a scalar. We define a scaled version of the random walk iterates, such that

$$\begin{aligned}\mathbf{y}^{(t)} &= (1-r)^{t/2}\mathbf{z}^{(t)}, \\ \mathbf{B}^{(t)} &= (1-r)^{t}\mathbf{C}^{(t)}, \text{ and} \\ \operatorname{Var}[\mathbf{y}_i^{(t)}] &= (1-r)^{t} \operatorname{Var}[\mathbf{z}_i^{(t)}]\end{aligned}$$

Because the sequence $\mathbf{z}^{(t)}$ converges to a non-zero stationary point, the scaled sequence $\mathbf{y}^{(t)}$ converges to zero with a linear rate r.

Lemma 5. Under the assumptions of Lemma 3, the variance $\operatorname{Var}[\mathbf{y}_i^{(t)}]$ is bounded as

$$\operatorname{Var}[\mathbf{y}_i^{(t)}] \leq \frac{(1-r)^t}{(1-\gamma)n_{\mathbf{W}}(\gamma)},$$

with equality as $t \to \infty$.

Proof. From Lemma 4, we know that $(1-r)^t \operatorname{Var}[\mathbf{z}_i^{(t)}] \leq (1-r)^t \lim_{t' \to \infty} \operatorname{Var}[\mathbf{z}_i^{(t')}]$, with equality as $t \to \infty$. Because the variance $\operatorname{Var}[\mathbf{z}_i^{(t)}]$ is equal across workers i (Lemma 3), the Lemma follows from rearranging Definition I. □

Lemma 6. The covariance vector $\mathbf{b}$ (the flattened version of $\mathbf{B}$) of this scaled random walk process follows the recursion $\mathbf{b}^{(t+1)} = \mathbf{T}^{\text{gossip}} u_t(\mathbf{b}^{(t)})$, where

$$u_t(\mathbf{b}^{(t)})_{ni+j} = \begin{cases} \gamma(1-r)\,\mathbf{b}_{ni+j}^{(t)} + (1-r)^{t+1} & i = j \text{ (same worker)}, \\ \gamma(1-r)\,\mathbf{b}_{ni+j}^{(t)} & i \neq j \text{ (different workers)}. \end{cases}$$

Proof. The entries $u_t(\mathbf{b}^{(t)})_{ni+j}$ are inner products:

$$\begin{aligned} u_t(\mathbf{b}^{(t)})_{ni+j} &= (1-r)^{t+1} \left\langle \sqrt{\gamma}\mathbf{y}_i^{(t)} + \boldsymbol{\xi}_i^{(t)}, \sqrt{\gamma}\mathbf{y}_j^{(t)} + \boldsymbol{\xi}_j^{(t)} \right\rangle \\ &= \gamma(1-r)\mathbf{b}_{ni+j}^{(t)} + (1-r)^{t+1}\,\mathbb{E}\left\langle \boldsymbol{\xi}_i^{(t)}, \boldsymbol{\xi}_j^{(t)} \right\rangle. \end{aligned}$$

The inner product between noise contributions $\boldsymbol{\xi}_i^{(t)}$ and $\boldsymbol{\xi}_j^{(t)}$ are 1 if $i = j$ and 0 otherwise. □

Lemma 7. The covariance $\mathbf{b}$ follows the recursion $\mathbf{b}^{(t+1)} \geq \mathbf{T}^{\text{gossip}}\mathbf{T}^{\text{r.w.}}\mathbf{b}^{(t)}$ (element-wise), where

$$[\mathbf{T}^{\text{r.w.}}\mathbf{b}^{(t)}]_{ni+j} = \begin{cases} \gamma(1-r)\,\mathbf{b}_{ni+j}^{(t)} + (1-r)(1-\gamma)n_{\mathbf{W}}(\gamma)\mathbf{b}_{ni+j}^{(t)} & i = j, \\ \gamma(1-r)\,\mathbf{b}_{ni+j}^{(t)} & i \neq j. \end{cases} \tag{16}$$

In the limit of $t \to \infty$, this is true with equality.

Proof. From Lemma 5, we have that $\mathbf{b}_{ni+i}^{(t)} = \operatorname{Var}[\mathbf{y}_i^{(t)}] \leq \frac{(1-r)^t}{(1-\gamma)n_{\mathbf{W}}(\gamma)}$, with equality as $t \to \infty$. The entries of $\mathbf{T}^{\text{r.w.}}\mathbf{b}^{(t)}$ are therefore all smaller than or equal to the entries of $u_t(\mathbf{b}^{(t)})$ from Lemma 6, which proves the Lemma. □

C.6 The rate for D-SGD

Theorem III (D-SGD on random quadratics)**.** Under assumptions D, E, F, and G, if the pair of the learning rate η and r satisfy

$$r = 1 - (1-\eta)^2 - \frac{(\zeta - 1)\eta^2}{n_{\mathbf{W}}\left(\frac{(1-\eta)^2}{1-r}\right)}, \tag{17}$$

the error of D-SGD with learning rate η on the random quadratic objective with noise parameter ζ converges with rate r:

$$\sum_{i=1}^{n} \mathbb{E}\|\mathbf{x}_i^{(t)}\|^2 \leq (1-r)^t \sum_{i=1}^{n} \mathbb{E}\|\mathbf{x}_i^{(0)}\|^2.$$

This rate becomes exact as $t \to \infty$.

Proof. If the condition (17) is satisfied, the expected error iterates $\mathbf{E}$ (Equation 13) of the D-SGD algorithm follow the transition matrix (16) of Lemma 7 with $\gamma = \frac{(1-\eta)^2}{1-r}$. The choice of γ ensures that $\gamma(1-r) = (1-\eta)^2$, and the condition (17) that $(\zeta-1)\eta^2 = (1-r)(1-\gamma)n_{\mathbf{W}}(\gamma)$.

From Lemma 7, we know that a sequence that has this transition matrix $\mathbf{T}^{\text{gossip}}\mathbf{T}^{\text{r.w.}}\mathbf{b}^{(t)}$ (16) converges at least as fast as the iterates of the corresponding scaled random walk process $\mathbf{B}$, with equality in the limit as $t \to \infty$.

Since $\mathbf{B}$ converges to zero with a rate r, this now implies the same rate for the error matrix $\mathbf{E}$. The sum $\sum_{i=1}^{n} \mathbb{E}\|\mathbf{x}_i^{(t)}\|^2$ in the statement of this theorem is the trace of the matrix $\mathbf{E}^{(t)}$, and therefore it converges to zero with the same rate. This completes the proof. □

D (Strongly)-Convex case, missing proofs and additional results

D.1 Preliminaries on Bregman divergences

Throughout this section, we will use Bregman divergences, which are defined for a differentiable function h and two points $x, y \in \mathbb{R}^d$ as:

$$D_h(x, y) = h(x) - h(y) - \nabla h(y)^\top (x - y). \tag{18}$$

We assume throughout this paper that the functions we consider are twice continuously differentiable and strictly convex on $\mathrm{dom}\, h$, and that $\nabla h(x) = \min_y h(y) - x^\top y$ is uniquely defined (although milder assumptions could be used). Among the many properties of these divergences, an important one is that if h is L smooth and μ strongly-convex, then

$$\frac{\mu}{2}\|x - y\|^2 \leq D_h(x, y) \leq \frac{L}{2}\|x - y\|^2 \tag{19}$$

Another important property is called duality, which states that:

$$D_h(x, y) = D_{h^*}(\nabla h(y), \nabla h(x)), \tag{20}$$

where h^* is the convex conjugate of h.

D.2 Main result

This section is devoted to proving Theorem IV, from which Theorem I can be deduced directly by taking $\omega = M_0$ and $p = 1/2$. We recall Assumption B, which is at the heart of Theorem IV.

Assumption B. The stochastic gradients are such that: (I) $\xi_i^{(t)}$ and $\xi_j^{(\ell)}$ are independent for all t, ℓ and $i \neq j$. (II) $\mathbb{E}\left[f_{\xi_i^{(t)}}\right] = f$ for all t, i (III) $\mathbb{E}\,\|\nabla f_{\xi_i^{(t)}}(\mathbf{x}^\star)\|^2 \leq \sigma^2$ for all t, i, where $\mathbf{x}^\star$ is a minimizer of f. (IV) $f_{\xi_i^{(t)}}$ is convex and ζ-smooth for all t, i. (V) f is μ-strongly-convex for $\mu \geq 0$ and L-smooth.

Note that Assumption B (IV) is stated in this form for simplicity, but it can be relaxed by asking directly that $\mathbb{E}\left[\|\nabla f_{\xi,i}(x^{(t)}) - \nabla f_{\xi,i}(x^\star)\|^2\right] \leq 2\zeta D_f(x^\star, x^{(t)})$, which can also be implied by assuming that each f_ξ is ζ_ξ-smooth, with $\mathbb{E}\left[\zeta_\xi D_{f_\xi}(x^\star, x^{(t)})\right] \leq \zeta D_f(x^\star, x^{(t)})$ (see Equation (28)). These weaker forms would be satisfied by the toy problem of Section 3.

Theorem IV. Denote $\mathbf{x}^{(t)}$ the iterates obtained by D-SGD, $\mathbf{L}_\mathbf{M} = \mathbf{I} - \mathbf{M}$, and p the probability to perform a communication step ($\mathbf{x}_{t+1} = \mathbf{W}\mathbf{x}_t$). Parameter β is such that $\mathbf{I} - \mathbf{W} \succcurlyeq \beta \mathbf{L}_\mathbf{M}$. For some $\omega > 0$, denote:

$$\mathcal{L}_t = \|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_\mathbf{M} + \omega \|\mathbf{x}^{(t)}\|^2_{\mathbf{L}_\mathbf{M}}. \tag{21}$$

Then, if η is such that:

$$\eta \leq \frac{M_0 \beta}{L}\frac{p}{1 - p}, \tag{22}$$

$$\eta \leq \frac{1}{4\left(M_0 \zeta + L\right)} \tag{23}$$

we have that:

$$\mathcal{L}_t \leq \left[1 - (1 - p)\eta\mu\right]^t \mathcal{L}_0 + \frac{\eta \tilde{\sigma}^2}{\mu}, \tag{24}$$

with $\tilde{\sigma}^2 = \sigma^2_\mathbf{M} + \omega \sigma^2_{\mathbf{L}_\mathbf{M}}$, where $\mathbb{E}\,\|\nabla f_{\xi_i^{(t)}}(\mathbf{x}^\star)\|^2_\mathbf{M} \leq \sigma^2_\mathbf{M}$ (and similarly for $\mathbf{L}_\mathbf{M}$).

In the convex case ($\mu = 0$), we have:

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} D_f(\mathbf{M}\mathbf{x}^{(t)}, \mathbf{x}^\star)\right] \leq \frac{1}{1 - p}\frac{\mathcal{L}_0}{\eta T} + \eta \tilde{\sigma}^2 \tag{25}$$

Note that the factors 2 in Equation (23) are simplifications to make the result more readable but could be improved.

Proof. We now proceed to the proof of the theorem. To show that the Lyapunov $\mathcal{L}_t$ decreases over iterations, we will study how each quantity $\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_\mathbf{M}$ and $\|\mathbf{x}^{(t)}\|^2_{\mathbf{L}_\mathbf{M}}$ evolves through time. In particular, we will first consider the case of computation updates (so, local gradient updates), and then the case of gossip updates.

1 - Computation updates In this case, we assume that the update is of the form

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \eta \nabla f_{\xi_t}(\mathbf{x}^{(t)}). \tag{26}$$

This happens with probability $1 - p$, and expectations are taken with respect to ξ_t. To avoid notation clutter, we use notations ∇f_ξ and $\nabla f_{\xi,i}$, which are such that $\nabla f_{\xi,i}(\mathbf{x}^{(t)}) = (\nabla f_\xi(\mathbf{x}^{(t)}))_i = \nabla f_{\xi_i^{(t)}}(\mathbf{x}_i^{(t)})$.

Distance to optimum We bound the distance to optimum as follows, using that $\mathbf{M}\mathbf{x}^\star = \mathbf{x}^\star$, and $\mathbb{E}\left[\nabla f_\xi(\mathbf{x}^{(t)})\right] = \nabla f(\mathbf{x}^{(t)})$:

$$\begin{aligned}
\mathbb{E}\left[\|\mathbf{x}^{(t+1)} - \mathbf{x}^\star\|^2_{\mathbf{M}}\right] &= \|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{M}} - 2\eta\, \mathbb{E}\left[(\mathbf{x}^{(t)} - \mathbf{x}^\star)^\top \mathbf{M} \nabla f_\xi(\mathbf{x}^{(t)})\right] + \eta^2 \|\nabla f_\xi(\mathbf{x}^{(t)})\|^2_{\mathbf{M}} \\
&= \|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{M}} - 2\eta(\mathbf{M}\mathbf{x}^{(t)} - \mathbf{x}^\star)^\top \nabla f(\mathbf{x}^{(t)}) + \eta^2\, \mathbb{E}\left[\|\nabla f_\xi(\mathbf{x}^{(t)})\|^2_{\mathbf{M}}\right].
\end{aligned}$$

Then, we expand the middle term in the following way:

$$\begin{aligned}
-\nabla f(\mathbf{x}^{(t)})^\top (\mathbf{M}\mathbf{x}^{(t)} - \mathbf{x}^\star) &= -\nabla f(\mathbf{x}^{(t)})^\top (\mathbf{x}^{(t)} - \mathbf{x}^\star) - \nabla f(\mathbf{x}^{(t)})^\top (\mathbf{M}\mathbf{x}^{(t)} - \mathbf{x}^{(t)}) \\
&= -D_f(\mathbf{x}^{(t)}, \mathbf{x}^\star) - D_f(\mathbf{x}^\star, \mathbf{x}^{(t)}) + D_f(\mathbf{M}\mathbf{x}^{(t)}, \mathbf{x}^{(t)}) - f(\mathbf{M}\mathbf{x}^{(t)}) + f(\mathbf{x}^{(t)}) \\
&= -D_f(\mathbf{M}\mathbf{x}^{(t)}, \mathbf{x}^\star) - D_f(\mathbf{x}^\star, \mathbf{x}^{(t)}) + D_f(\mathbf{M}\mathbf{x}^{(t)}, \mathbf{x}^{(t)}) \\
&\leq -\frac{\mu}{2}\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{M}^2} - D_f(\mathbf{x}^\star, \mathbf{x}^{(t)}) + \frac{L}{2}\|\mathbf{M}\mathbf{x}^{(t)} - \mathbf{x}^{(t)}\|^2, \qquad (27)
\end{aligned}$$

where in the last time we used the μ-strong convexity and L-smoothness of f. For the noise term, we use that fact that $(\nabla f_\xi(\mathbf{x}^{(t)}))_i$ and $(\nabla f_\xi(\mathbf{x}^{(t)}))_j$ are independent for $i \neq j$, so that

$$\begin{aligned}
\frac{1}{2}\mathbb{E}\left[\|\nabla f_\xi(\mathbf{x}^{(t)})\|^2_{\mathbf{M}}\right] &= \mathbb{E}\left[\|\nabla f_\xi(\mathbf{x}^{(t)}) - \nabla f_\xi(\mathbf{x}^\star)\|^2_{\mathbf{M}}\right] + \mathbb{E}\left[\|\nabla f_\xi(\mathbf{x}^\star)\|^2_{\mathbf{M}}\right] \\
&= \sum_{i=1}^n \mathbf{M}_{ii}\, \mathbb{E}\left[\|\nabla f_{\xi,i}(\mathbf{x}^{(t)}) - \nabla f_{\xi,i}(\mathbf{x}^\star)\|^2\right] + \mathbb{E}\left[\|\nabla f_\xi(\mathbf{x}^\star)\|^2_{\mathbf{M}}\right] \\
&\quad + \sum_{i=1}^n \sum_{j\neq i} \mathbf{M}_{ij}\, \mathbb{E}\left[[\nabla f_{\xi,i}(\mathbf{x}^{(t)}) - \nabla f_{\xi,i}(\mathbf{x}^\star)]^\top [\nabla f_{\xi,j}(\mathbf{x}^{(t)}) - \nabla f_{\xi,j}(\mathbf{x}^\star)]\right] \\
&= \sum_{i=1}^n \mathbf{M}_{ii}\, \mathbb{E}\left[\|\nabla f_{\xi,i}(\mathbf{x}^{(t)}) - \nabla f_{\xi,i}(\mathbf{x}^\star)\|^2\right] + \|\nabla f(\mathbf{x}^{(t)})\|^2_{\mathbf{M}} + \mathbb{E}\left[\|\nabla f_\xi(\mathbf{x}^\star)\|^2_{\mathbf{M}}\right].
\end{aligned}$$

We now use for all $i \in \{1, \dots, n\}$ the ζ-smoothness of $f_{\xi,i}$, which implies the ζ^{-1}-strong convexity of $f^*_{\xi,i}$ [7], so that:

$$\begin{aligned}
\mathbb{E}\left[\|\nabla f_{\xi,i}(\mathbf{x}^{(t)}) - \nabla f_{\xi,i}(\mathbf{x}^\star)\|^2\right] &= 2\,\mathbb{E}\left[D_{\frac{1}{2}\|\cdot\|^2}(\nabla f_{\xi,i}(\mathbf{x}^{(t)}), \nabla f_{\xi,i}(\mathbf{x}^\star))\right] \\
&\leq \frac{2}{\zeta^{-1}}\,\mathbb{E}\left[D_{f^*_{\xi,i}}(\nabla f_{\xi,i}(\mathbf{x}^{(t)}), \nabla f_{\xi,i}(\mathbf{x}^\star))\right] \\
&\leq 2\zeta\,\mathbb{E}\left[D_{f_{\xi,i}}(\mathbf{x}^\star, \mathbf{x}^{(t)})\right] \qquad (28) \\
&= 2\zeta D_f(\mathbf{x}^\star, \mathbf{x}^{(t)})
\end{aligned}$$

For the expected gradient term, we can use that:

$$\|\nabla f(\mathbf{x}^{(t)})\|^2_{\mathbf{M}} \leq \|\nabla f(\mathbf{x}^{(t)}) - \nabla f(\mathbf{x}^\star)\|^2 \leq 2L D_f(\mathbf{x}^\star, \mathbf{x}^{(t)}),$$

so that in the end,

$$\mathbb{E}\left[\|\nabla f_\xi(\mathbf{x}^{(t)})\|^2_{\mathbf{M}}\right] \leq 4(\zeta M_0 + L) D_f(\mathbf{x}^\star, \mathbf{x}^{(t)}) + 2\sigma^2_{\mathbf{M}}, \tag{29}$$

where $M_0 = \max_i \mathbf{M}_{ii}$, and $\mathbb{E}\left[\|\nabla f_\xi(\mathbf{x}^\star)\|^2_{\mathbf{M}}\right] \leq \sigma^2_{\mathbf{M}}$, the locally averaged variance at optimum. Plugging this into the main equation, we obtain that:

$$\begin{aligned}
\mathbb{E}\left[\|\mathbf{x}^{(t+1)} - \mathbf{x}^\star\|^2_{\mathbf{M}}\right] &\leq \|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{M}} - \eta\mu\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{M}^2} + 2\eta^2\sigma^2_{\mathbf{M}} \\
&\quad - 2\eta\left(1 - 2\eta\left[\zeta M_0 + L\right]\right) D_f(\mathbf{x}^\star, \mathbf{x}^{(t)}) + \frac{L}{2}\|\mathbf{M}\mathbf{x}^{(t)} - \mathbf{x}^{(t)}\|^2
\end{aligned}$$

The last step is to write that $\mathbf{M}^2 = \mathbf{M} - \mathbf{M}\mathbf{L}_\mathbf{M}$, so that:

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{x}^{(t+1)} - \mathbf{x}^\star\|^2_\mathbf{M}\right] \leq (1 - \eta\mu)\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_\mathbf{M} + \eta\mu\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{M}\mathbf{L}_\mathbf{M}} + 2\eta^2\sigma^2_\mathbf{M} \\ - 2\eta\left(1 - 2\eta\left[\zeta M_0 + L\right]\right) D_f(\mathbf{x}^\star, \mathbf{x}^{(t)}) + \eta L\|\mathbf{M}\mathbf{x}^{(t)} - \mathbf{x}^{(t)}\|^2\end{aligned}$$

At this point, we can use that $\mathbf{M}\mathbf{L}_\mathbf{M} \leq \mathbf{I}/4$,

$$\mu\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{M}\mathbf{L}_\mathbf{M}} \leq \frac{\mu}{4}\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2 \leq \frac{1}{2}D_f(\mathbf{x}^\star, \mathbf{x}^{(t)}), \tag{30}$$

so that

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{x}^{(t+1)} - \mathbf{x}^\star\|^2_\mathbf{M}\right] \leq (1 - \eta\mu)\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_\mathbf{M} + \eta L\|\mathbf{M}\mathbf{x}^{(t)} - \mathbf{x}^{(t)}\|^2 \\ - 2\eta\left(3/4 - 2\eta\left[\zeta M_0 + L\right]\right) D_f(\mathbf{x}^\star, \mathbf{x}^{(t)}) + 2\eta^2\sigma^2_\mathbf{M}\end{aligned} \tag{31}$$

Distance to consensus We now bound the distance to consensus in the case of a communication update. More specifically, we write that:

$$\mathbb{E}\left[\|\mathbf{x}^{(t+1)}\|^2_{\mathbf{L}_\mathbf{M}}\right] = \|\mathbf{x}^{(t)}\|^2_{\mathbf{L}_\mathbf{M}} - 2\eta\nabla f(\mathbf{x}^{(t)})^\top\mathbf{L}_\mathbf{M}\mathbf{x}^{(t)} + \eta^2\,\mathbb{E}\left[\|\nabla f_{\xi_t}(\mathbf{x}^{(t)})\|^2_{\mathbf{L}_\mathbf{M}}\right]$$

Then, we develop the middle term as:

$$\begin{aligned}-\nabla f(\mathbf{x}^{(t)})^\top\mathbf{L}_\mathbf{M}\mathbf{x}^{(t)} &= -\nabla f(\mathbf{x}^{(t)})^\top(\mathbf{I} - \mathbf{M})\mathbf{x}^{(t)} \\ &= \nabla f(\mathbf{x}^{(t)})^\top(\mathbf{M}\mathbf{x}^{(t)} - \mathbf{x}^{(t)}) \\ &= -D_f(\mathbf{M}\mathbf{x}^{(t)}, \mathbf{x}^{(t)}) + f(\mathbf{M}\mathbf{x}^{(t)}) - f(\mathbf{x}^{(t)}) \\ &\leq -\frac{\mu}{2}\|\mathbf{M}\mathbf{x}^{(t)} - \mathbf{x}^{(t)}\|^2 + f(\mathbf{M}\mathbf{x}^{(t)}) - f(\mathbf{x}^{(t)})\end{aligned}$$

By convexity of f (since the expected function is the same for all workers), we have that

$$f(\mathbf{M}\mathbf{x}^{(t)}) \leq f(\mathbf{x}^{(t)}). \tag{32}$$

We finally decompose ${\mathbf{L}_\mathbf{M}}^2 = \mathbf{L}_\mathbf{M}(\mathbf{I} - \mathbf{M})$, so that:

$$-2\eta\nabla f(\mathbf{x}^{(t)})^\top\mathbf{L}_\mathbf{M}\mathbf{x}^{(t)} \leq -\eta\mu\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{L}_\mathbf{M}} + \eta\mu\|\mathbf{x}^{(t)}\|^2_{\mathbf{M}\mathbf{L}_\mathbf{M}}$$

For the noise term, we obtain exactly the same derivations as in the previous setting, but this time with matrix $\mathbf{L}_\mathbf{M} = \mathbf{I} - \mathbf{M}$ instead. Using the same bounding, and $M_{\min} = \min_i(\mathbf{M})_{ii}$, we thus obtain:

$$\mathbb{E}\left[\|\nabla f_\xi(\mathbf{x}^{(t)})\|^2_{\mathbf{L}_\mathbf{M}}\right] \leq 4(\zeta(1 - M_{\min}) + L)D_f(\mathbf{x}^\star, \mathbf{x}^{(t)}) + 2\sigma^2_{\mathbf{L}_\mathbf{M}}. \tag{33}$$

In particular, we have that:

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{x}^{(t+1)} - \mathbf{x}^\star\|^2_{\mathbf{L}_\mathbf{M}}\right] \leq (1 - \eta\mu)\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{L}_\mathbf{M}} + \eta\mu\|\mathbf{x}^{(t)}\|^2_{\mathbf{M}\mathbf{L}_\mathbf{M}} + 2\eta^2\sigma^2_{\mathbf{L}_\mathbf{M}} \\ + 4\eta^2(\zeta(1 - M_{\min}) + L)D_f(\mathbf{x}^\star, \mathbf{x}^{(t)}).\end{aligned}$$

Similarly to before, we use that

$$\mu\|\mathbf{x}^{(t)}\|^2_{\mathbf{M}\mathbf{L}_\mathbf{M}} = \mu\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{M}\mathbf{L}_\mathbf{M}} \leq \frac{\mu}{4}\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2 \leq \frac{1}{2}D_f(\mathbf{x}^\star, \mathbf{x}^{(t)}), \tag{34}$$

so that for computation updates, the distance to consensus evolves as:

$$\mathbb{E}\left[\|\mathbf{x}^{(t+1)} - \mathbf{x}^\star\|^2_{\mathbf{L}_\mathbf{M}}\right] \leq (1-\eta\mu)\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{L}_\mathbf{M}} + 2\eta\left[\frac{1}{4} + 2\eta(\zeta(1 - M_{\min}) + L)\right]D_f(\mathbf{x}^\star, \mathbf{x}^{(t)}) + 2\eta^2\sigma^2_{\mathbf{L}_\mathbf{M}} \tag{35}$$

Combining Equation (35) with Equation (31) leads to:

$$\begin{aligned}\mathcal{L}^{(t+1)} \leq (1 - \eta\mu)\mathcal{L}_t + \eta L\|\mathbf{M}\mathbf{x}^{(t)} - \mathbf{x}^{(t)}\|^2 + 2\eta^2\tilde{\sigma}^2 \\ - \eta\left(1 - 4\eta\left[\zeta(M_0 + \omega(1 - M_{\min})) + (1 + \omega)L\right]\right) D_f(\mathbf{x}^\star, \mathbf{x}^{(t)}),\end{aligned} \tag{36}$$

with $\tilde{\sigma}^2 = \sigma^2_\mathbf{M} + \omega\sigma^2_{\mathbf{L}_\mathbf{M}}$.

2 - Communication updates We write:

$$\begin{aligned}\|\mathbf{x}^{(t+1)} - \mathbf{x}^\star\|^2_{\mathbf{L_M}} &= \|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{W L_M W}}\\ &\leq \|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{W L_M}}\\ &= \|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{L_M}} - \|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{L_W L_M}}\end{aligned}$$

For distance to optimum part in communication update, we obtain:

$$\|\mathbf{x}^{(t+1)} - \mathbf{x}^\star\|^2_{\mathbf{M}} = \|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{WMW}} \leq \|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{M}} \tag{37}$$

We now introduce β, the strong convexity of $\mathbf{L_W} = \mathbf{I} - \mathbf{W}$ relative to $\mathbf{L_M}$:

$$\mathbf{L_W} \geq \beta \mathbf{L_M}. \tag{38}$$

Therefore, we obtain that for communication updates,

$$\mathcal{L}^{(t+1)} \leq \mathcal{L}_t - \omega\beta\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{L_M}^2}. \tag{39}$$

Putting terms back together We now put everything together, assuming that communication steps happen with probability p (and so computations steps with probability $1-p$). Thus, we mix Equations (36) and (39) to obtain:

$$\begin{aligned}\mathbb{E}\left[\mathcal{L}^{(t+1)}\right] \leq\ & (1-(1-p)\eta\mu)\mathcal{L}_t + 2(1-p)\eta^2\tilde{\sigma}^2\\ &+ \left[(1-p)\eta L - \omega p\beta\right]\|\mathbf{x}^{(t)}\|^2_{\mathbf{L_M}^2}\\ &- \eta(1-p)\left(1 - 4\eta\left[\zeta(M_0 + \omega(1-M_{\min})) + (1+\omega)L\right]\right) D_f(\mathbf{x}^\star, \mathbf{x}^{(t)}).\end{aligned}$$

In particular, we obtain the linear decrease of the Lyapunov $\mathcal{L}_t$ under the following conditions:

$$\begin{aligned}\eta &\leq \frac{\omega\beta}{L}\frac{p}{1-p}\\ \eta &\leq \frac{1}{4\left(\zeta\left[M_0 + \omega(1-M_{\min})\right] + (1+\omega)L\right)}\end{aligned}$$

Under these conditions, we have that

$$\mathbb{E}\left[\mathcal{L}^{(t+1)}\right] \leq (1-(1-p)\eta\mu)\mathcal{L}_t + (1-p)\eta^2\tilde{\sigma}^2, \tag{40}$$

and we can simply chain this relation to finish the proof of the theorem. □

Convex case In the convex case ($\mu = 0$), the proof is very similar, except that we keep the $D_f(\mathbf{M}\mathbf{x}^{(t)}, \mathbf{x}^\star)$ term from Equation (27). In particular, under the same step-size conditions as the strongly convex case, this leads to:

$$\mathbb{E}\left[\mathcal{L}^{(t+1)}\right] \leq \mathcal{L}_t + 2(1-p)\eta^2\tilde{\sigma}^2 - \eta(1-p)D_f(\mathbf{M}\mathbf{x}^{(t)}, \mathbf{x}^\star). \tag{41}$$

This leads to:

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} D_f(M\mathbf{x}^{(t)}, \mathbf{x}^\star)\right] \leq \frac{1}{1-p}\frac{\mathcal{L}_0}{\eta T} + 2\eta\tilde{\sigma}^2, \tag{42}$$

which finishes the proof of the theorem.

Evaluating β There are two important graph quantities: M_0 and β. If we choose M as in Equation (9), then its eigenvalues are equal to $\frac{(1-\gamma)\lambda_i^2}{1-\gamma\lambda_i^2}$, where λ_i is the i-th eigenvalue of $\mathbf{W}$. Therefore,

$$\lambda_i^{\mathbf{L_M}} = \frac{1-\lambda_i^2}{1-\gamma\lambda_i^2}. \tag{43}$$

In particular, we have that for all i,

$$1 - \lambda_i \geq \beta\frac{1-\lambda_i^2}{1-\gamma\lambda_i^2}, \tag{44}$$

so that we can take

$$\beta = \frac{1-\gamma{\lambda_2}^2}{1+\lambda_2} \geq \frac{1-\gamma\lambda_2}{2}, \tag{45}$$

where we use $\lambda_2 \leq 1$ to simplify the results. In particular, β does not depend on the spectral gap of $\mathbf{W}$ (which is equal to $1-\lambda_2$) as long as γ is not too large. Yet, an interesting phenomenon happens: *a larger graph also implies more effective neighbors for a given γ.*

Choice of ω A reasonable value for ω is to simply take it as $\omega = M_0$. Indeed,

- The second condition almost does not benefit from $\omega \leq M_0$ (factor 2 at most).
- If the first condition dominates, such that taking $\omega \geq M_0$ would loosen it, then instead one can reduce γ. This will lead to a higher value for both M_0 (and so for ω) and β. Note that, again, increasing M_0 does not make the second condition stronger than what it would have been with just increasing ω by more than a factor 2.

With this choice, we thus obtain that:

$$\eta \leq \min\left(\frac{M_0\beta}{L}\frac{p}{1-p}, \frac{1}{4\left(M_0\zeta(2-M_{\min}) + (1+M_0)L\right)}\right), \tag{46}$$

and Theorem IV is obtained by taking $M_0 \leq 1$ and $M_{\min} \geq 0$.

D.3 Obtaining Corollary II

In this section, we discuss the derivations leading to Corollary II. To do so, we start by making the simplifying assumption that

$$\frac{\zeta}{n} \geq L. \tag{47}$$

Using this, and writing $n_{\mathbf{W}}(\gamma) = 1/M_0$, the condition from Equation (23) simplifies to:

$$\eta \leq \frac{Ln_{\mathbf{W}}(\gamma)}{16\zeta}. \tag{48}$$

We always want this condition to be tight, and not Equation (22) the communication one, which is only there to allow us to use larger values of $n_{\mathbf{W}}(\gamma)$. In particular, we want that:

$$\frac{Ln_{\mathbf{W}}(\gamma)}{16\zeta} \leq \frac{\beta}{n_{\mathbf{W}}(\gamma)L}. \tag{49}$$

When we increase γ, $n_{\mathbf{W}}(\gamma)$ increases and β decreases. We thus want to take the highest γ such that (49) is verified (potentially with an equality if $n_{\mathbf{W}}(\gamma) < n$).

D.4 Deterministic algorithm

So far, we have analyzed the randomized variant of D-SGD, in which at each step, there is a coin flip to decide whether to perform a communication or computation step. We now show how to extend the analysis to the case in which:

$$\mathbf{x}^{(t+1)} = \mathbf{W}\mathbf{x}^{(t)} - \eta\nabla f_\xi(\mathbf{W}\mathbf{x}^{(t)}) \tag{50}$$

Note that D-SGD is often presented as $\mathbf{x}_{t+1} = \mathbf{W}(\mathbf{x}_t - \eta\nabla f_\xi(\mathbf{x}_t))$, but it turns out that the analysis is easier when considering it in the form of Equation (50). Yet, it comes down to the same algorithm (alternating communication and computation steps), and the difference simply is whether the error is evaluated after a communication step or a local gradient step. The results in the previous section did not depend on the value of $\mathbf{x}_t$, so we can perform the same derivations with $\mathbf{W}\mathbf{x}_t$ instead of $\mathbf{x}_t$, so that Equation (36) now writes:

$$\begin{aligned}\mathcal{L}(\mathbf{x}^{(t+1)}) &= (1-\eta\mu)\mathcal{L}(\mathbf{W}\mathbf{x}^{(t)}) + \eta L\|\mathbf{M}\mathbf{W}\mathbf{x}^{(t)} - \mathbf{W}\mathbf{x}^{(t)}\|^2 + 2\eta^2\tilde{\sigma}^2 \\ &\quad - \eta\left(1 - 4\eta\left[\zeta(M_0 + \omega(1-M_{\min})) + (1+\omega)L\right]\right) D_f(\mathbf{x}^\star, \mathbf{W}\mathbf{x}^{(t)}),\end{aligned} \tag{51}$$

where $\mathcal{L}(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}^\star\|^2_{\mathbf{M}} + \omega\|\mathbf{x}\|^2_{\mathbf{L_M}}$, so that $\mathcal{L}^{(t)} = \mathcal{L}(\mathbf{x}^{(t)})$. In particular, choosing η such that the second line is always negative (as before) leads to:

$$\mathcal{L}(\mathbf{x}^{(t+1)}) = (1-\eta\mu)\mathcal{L}(\mathbf{W}\mathbf{x}^{(t)}) + \eta L\|\mathbf{x}^{(t)}\|^2_{\mathbf{W}\mathbf{L_M}^2\mathbf{W}} + 2\eta^2\tilde{\sigma}^2. \tag{52}$$

Similarly, using Equation (39), we obtain that

$$\mathcal{L}(\mathbf{W}\mathbf{x}^{(t)}) \leq \mathcal{L}(\mathbf{x}^{(t)}) - \omega\beta\|\mathbf{x}^{(t)} - \mathbf{x}^\star\|^2_{\mathbf{L_M}^2}. \tag{53}$$

Combining Equations (52) and (53) and using that $\mathbf{W}\mathbf{L_M}^2\mathbf{W} \preccurlyeq \mathbf{L_M}^2$, we obtain:

$$\mathcal{L}^{(t+1)} \leq (1-\eta\mu)\mathcal{L}^{(t)} + (\eta L - (1-\eta\mu)\omega\beta)\|\mathbf{x}^{(t)}\|^2_{\mathbf{L_M}^2} + 2\eta^2\tilde{\sigma}^2. \tag{54}$$

Thus, we obtain similar guarantees (up to a factor $1-\eta\mu$ which is small) for the deterministic and randomized algorithms. Note that in this case, constant β can be replaced by a slightly better constant $\tilde{\beta}$ which would be such that:

$$\mathbf{L_M}\mathbf{L_W} \geq \tilde{\beta}\,\mathbf{W}\mathbf{L_M}^2\mathbf{W}. \tag{55}$$

E Cifar-10 experimental setup

Table 2 describes the details of our experiments with D-SGD with VGG-11 on Cifar-10.

Table 2: Default experimental settings for Cifar-10/VGG-11

Dataset	Cifar-10 [9]
Data augmentation	Random horizontal flip and random 32×32 cropping
Data normalization	Subtract mean $(0.4914, 0.4822, 0.4465)$ and divide standard deviation $(0.2023, 0.1994, 0.2010)$
Architecture	VGG-11 [19]
Training objective	Cross entropy
Evaluation objective	Top-1 accuracy
Number of workers	32 (unless otherwise specified)
Topology	Ring (unless otherwise specified)
Gossip weights	Metropolis-Hastings (1/3 for ring, $w_{ij} = 1/(\max(n_i, n_j) + 1)$, worker i has n_i direct neighbors)
Data distribution	Identical: workers can sample from the whole dataset
Sampling	With replacement (i.i.d.), *no* shuffled passes
Batch size	16 patches per worker
Momentum	0.9 (heavy ball / PyTorch default)
Learning rate	Exponential grid or tuned for lowest training loss after 25 epochs
LR decay	Step-wise, $\times 0.1$ at epoch 75% and 90% of training
LR warmup	None
# Epochs	100 (full training) or only 25 (initial phase), based on total number of gradient accesses across workers
Weight decay	10^{-4}
Normalization scheme	no normalization layers
Exponential moving average	$\mathbf{x}_{\text{ema}}^{(t)} = 0.95\mathbf{x}_{\text{ema}}^{(t-1)} + 0.05\mathbf{x}^{(t)}$. This influences evaluation, not training
Repetitions per training	Just 1 per learning rate, but experiments are very consistent across similar learning rates
Reported metrics	*Loss after 2.5 k steps*: to reduce noise, we take two measures: (I) we use exponential moving average of the model parameters, and (II) we fit a parametric model $\log(l) = at + b$ to the 25 loss evaluations (t, l) closest to $t = 2500$. We then evaluate this function at $t = 2500$.

F Additional experiments

In the main paper, we have focussed on the training loss in the initial phase of training of Cifar-10. We do find that our findings there do correlate with test accuracy after a complete training with 100 epochs. Figure 10 shows the test accuracy as training progresses, for plots ordered by improving training loss after 2.5 k steps.

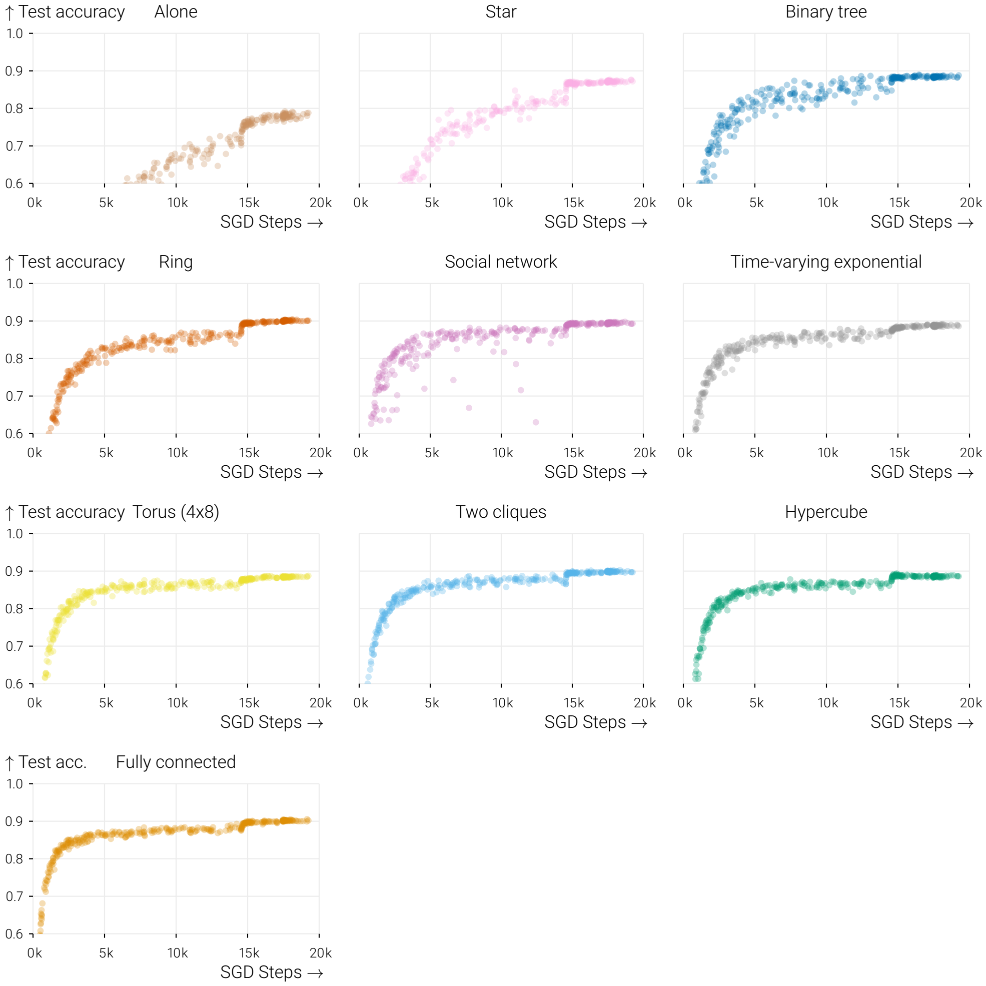

Figure 10: Test accuracy over the course of training a VGG-11 network on Cifar-10. See Appendix E for all details on the experimental setup. The plots are ordered by improving training loss after 2.5 k SGD steps. This ordering correlates well with the speed of improvements in test accuracy.

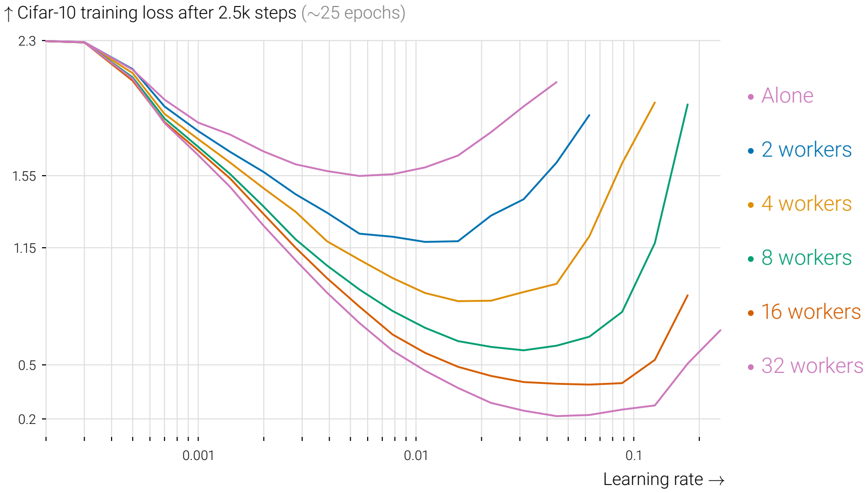

Figure 11: Training loss reached after 2.5 k SGD steps with fully-connected topologies of varying size. Averaging with more workers speeds up convergence for fixed learning rates, but also allows larger learning rates to be used. This plot serves as a reference for Figure 4, which shows similar plots for a variety of graph topologies.

F.1 Results on Fashion MNIST

We replicated our main experiments (Cifar-10/VGG) on another dataset and another network architecture. We chose for the Fashion MNIST dataset [24] and a simple multi-layer perceptron architecture with one hidden layer of 5000 neurons and ReLU activations. We list the details of our experimental setup in Table 3. We varied two key parameters compared to our Cifar-10 results: we used 64 workers instead of 32, and used SGD *without* momentum and *without* weight decay. Because this task is easier than Cifar-10, the initial phase where both training and test loss converge at similar rates is shorter. We therefore consider the first 500 steps as the 'initial phase', as opposed to 2500.

Figures 12 and 13 correspond to figures 4 and 6 from the main paper. We find that the conclusions from the paper also hold in this different experimental setting.

Table 3: Experimental settings for Fashion MNIST. Differences with Cifar-10 in **red**.

Dataset	**Fashion MNIST** [24]
Data augmentation	**None**
Data normalization	Subtract mean 0.2860 and divide standard deviation 0.3530
Architecture	**MLP** ($28 \times 28 \to \text{ReLU} \to 5000 \to \text{ReLU} \to 10$)
Training objective	Cross entropy
Evaluation objective	Top-1 accuracy
Number of workers	**64** (unless otherwise specified)
Topology	Ring (unless otherwise specified)
Gossip weights	Metropolis-Hastings (1/3 for ring, $w_{ij} = 1/(\max(n_i, n_j) + 1)$, worker i has n_i direct neighbors)
Data distribution	Identical: workers can sample from the whole dataset
Sampling	With replacement (i.i.d.), *no* shuffled passes
Batch size	16 patches per worker
Momentum	**0.0**
Learning rate	Exponential grid or tuned for lowest training loss after **500 steps**
LR decay	None, in the initial phase of training
LR warmup	None
# Epochs	500 steps
Weight decay	**0**
Normalization scheme	no normalization layers
Exponential moving average	$\mathbf{x}_{\text{ema}}^{(t)} = 0.95\mathbf{x}_{\text{ema}}^{(t-1)} + 0.05\mathbf{x}^{(t)}$. This influences evaluation, not training
Repetitions per training	Just 1 per learning rate, but experiments are very consistent across similar learning rates
Reported metrics	*Loss after **500** steps*: to reduce noise, we take two measures: (I) we use exponential moving average of the model parameters, and (II) we fit a parametric model $\log(l) = at + b$ to the 25 loss evaluations (t, l) closest to $t = 500$. We then evaluate this function at $t = 500$.

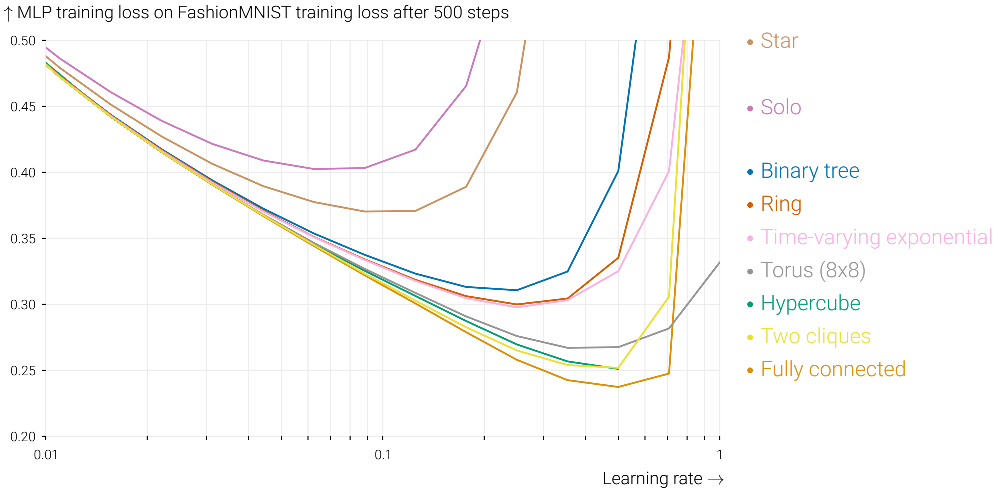

Figure 12: Training loss reached after 500 SGD steps with a variety of 64-worker graph topologies. In all cases, averaging yields a small increase in speed for small learning rates, but a large gain over training alone comes from being able to increase the learning rate. While the star has a better spectral gap (0.0156) than the ring (0.0032), it performs worse, and does not allow large learning rates.

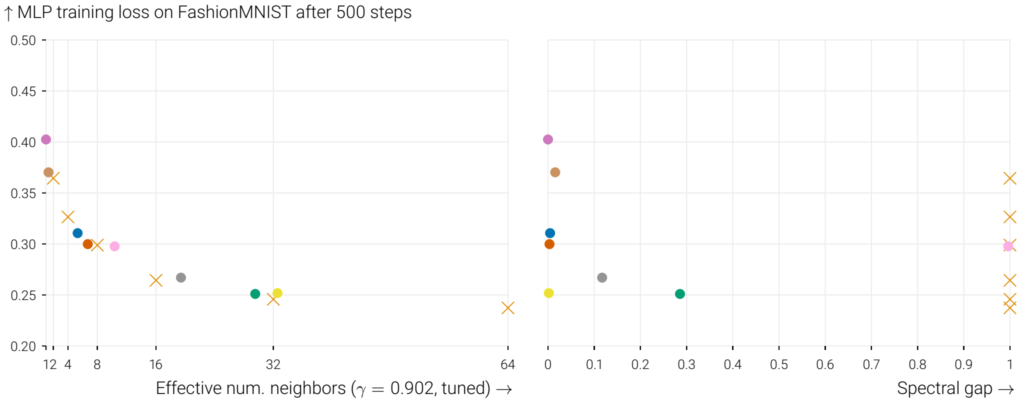

Figure 13: Fashion MNIST training loss after 500 steps for all studied topologies with their optimal learning rates. Colors match Figure 12, and × indicates fully-connected graphs with varying number of workers. After fitting a decay parameter $\gamma = 0.902$ that captures problem specifics, the effective number of neighbors (left) as measured by variance reduction in a random walk (like in Section 3) explains the relative performance of these graphs much better than the spectral gap of these topologies (right).

F.2 Heterogeneous data

While our experimental and theoretical data only describe the setting in which workers optimize objectives with a shared optimum, we believe that our insights are meaningful for heterogeneous settings as well. With heterogeneous data, we observe two regimes: in the beginning of training, when the worker's distant optima are in a similar direction, everything behaves identical to the homogeneous setting. In this regime, our insights seem to apply directly. Heterogeneity only plays a role later during the training, when it leads to conflicting gradient directions. This behavior is illustrated on a toy problem in Figure 14. We run D-SGD on our isotropic quadratic toy problem ($d = 100$, $n = 32$), but where the optima are removed from zero as a normal distribution with standard deviations 0, 10^{-7}, and 10^{-3} respectively. The (constant) learning rates are tuned for each topology in the homogeneous setting.

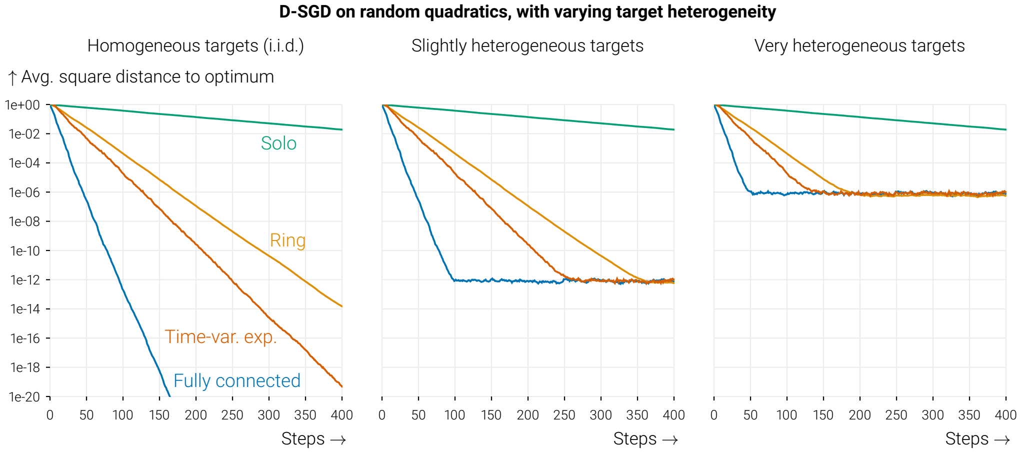

Figure 14: Convergence curves on our isotropic random quadratics problem (Section 3, with $d = 100$, $n = 32$), but where the optima are removed from zero as a zero-mean normal distribution with standard deviations 0, 10^{-7}, and 10^{-3} respectively. Constant learning rates are tuned independently for each topology in the homogeneous setting. Heterogeneity does not affect the initial phase of training, and our insights about maximum learning rates and the quality of communication topologies hold in this regime.

F.3 The role of γ in the experiments

In Figure 5, we optimize γ independently for each topology, minimizing the Mean Squared Error between the normalized covariance matrix measured from checkpoints of Cifar-10 training and the covariance in a random walk with the decay parameter γ. The bottom two rows of Figure 15 below show how Figure 5 would change, if you used a γ that is either much too low, or too high.

In Figure 6, we choose a value of γ (shared between all topologies) that yields a good correspondence between the performance of fully connected topologies (with 2, 4, 8, 16 and 32 workers) and the other topologies. We opt for sharing a single γ here, to test whether this metric could have predictive power for the quality of graphs. Figure 16 below shows how the figure changes if you use a value of γ that is either much too low, or much too high.

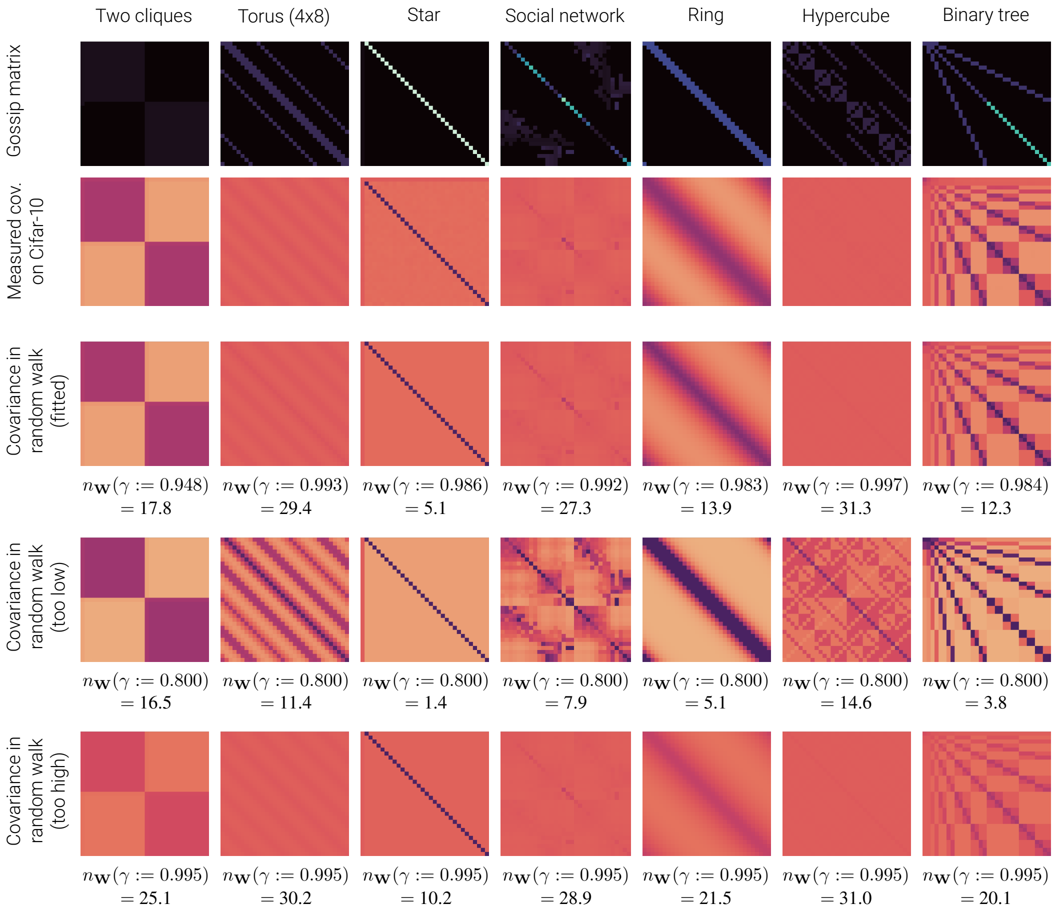

Figure 15: Extension of Figure 5. Measured covariance in Cifar-10 (second row) between workers using various graphs (top row). After 10 epochs, we store a checkpoint of the model and train repeatedly for 100 SGD steps, yielding 100 models for 32 workers. We show normalized covariance matrices between the workers. These are very well approximated by the covariance in the random walk process of Section 3 (third row). We print the fitted decay parameters and corresponding 'effective number of neighbors'. The bottom two rows show how Figure 5 would change, if you used a γ that is either much too low, or too high.

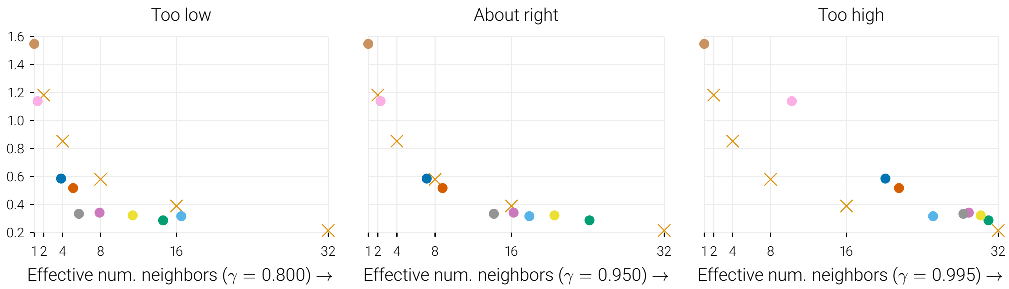

Figure 16: Extension of Figure 6, demonstrating how the fit changes if you use a value of γ that is either too low (left) or too high (right).