Consensus Control for Decentralized Deep Learning

Lingjing Kong [1 *] **Tao Lin** [1 *] **Anastasia Koloskova** [1] **Martin Jaggi** [1] **Sebastian U. Stich** [1]

Abstract

Decentralized training of deep learning models enables on-device learning over networks, as well as efficient scaling to large compute clusters. Experiments in earlier works reveal that, even in a data-center setup, decentralized training often suffers from the degradation in the quality of the model: the training and test performance of models trained in a decentralized fashion is in general worse than that of models trained in a centralized fashion, and this performance drop is impacted by parameters such as network size, communication topology and data partitioning.

We identify the changing consensus distance between devices as a key parameter to explain the gap between centralized and decentralized training. We show in theory that when the training consensus distance is lower than a critical quantity, decentralized training converges as fast as the centralized counterpart. We empirically validate that the relation between generalization performance and consensus distance is consistent with this theoretical observation. Our empirical insights allow the principled design of better decentralized training schemes that mitigate the performance drop. To this end, we provide practical training guidelines and exemplify its effectiveness on the data-center setup as the important first step.

1. Introduction

The impressive successes of machine learning, witnessed in the last decade, have been accompanied by a steady increase in the size, complexity, and computational requirements of training systems. In response to these challenges, distributed training algorithms (i.e. data-parallel large mini-batch SGD) have been developed for the use in data-centers (Goyal et al., 2017; You et al., 2018; Shallue et al., 2018). These state-of-the-art (SOTA) training systems rely on the All-Reduce communication primitive to perform exact averaging on the local mini-batch gradients computed on different subsets of the data, for the later synchronized model update. However, exact averaging with All-Reduce is sensitive to the communication hardware of the training system, causing the bottleneck in efficient deep learning training. To address this issue, decentralized training has become an indispensable training paradigm for efficient large scale training in data-centers (Assran et al., 2019), alongside its orthogonal benefits on preserving users' privacy for edge AI (Bellet et al., 2018; Kairouz et al., 2019).

Decentralized SGD (D-SGD) implementations trade off the exactness of the averaging provided by All-Reduce, with more efficient, but inexact, communication over sparse typologies. However, this often results in a severe drop in the training and/or test performance (i.e. generalization gap), even after hyper-parameter fine-tuning (see our Table 1 as well as Tables 1–3 in Assran et al., 2019). This phenomenon is poorly understood even in relatively straightforward i.i.d. data distribution scenarios (i.e. the data-center case), to which very few works are dedicated (in fact none of them provide insights into the generalization performance).

Table 1: **Significant generalization gap for decentralized training** on a sparse ring topology (ResNet-20 on CIFAR-10 with $n \in \{16, 32, 64\}$ workers). Decentralized SGD (D-SGD) communicates model parameters through the gossip averaging. Test top-1 accuracies averaged over three seeds with fine-tuned learning rates.

	AllReduce (complete)	D-SGD (ring)
n=16	92.91 ± 0.12	92.40 ± 0.10
n=32	92.82 ± 0.27	91.81 ± 0.09
n=64	92.71 ± 0.11	89.58 ± 0.20

In this work, we investigate the trade-off between the train/test performance and the exactness of the averaging, measured in terms of consensus distance, i.e. the average discrepancy between each node and the mean of model parameters over all machines. We identify this consensus distance as the key parameter that captures the joint effect of decentralization.

While one might suspect that a smaller consensus distance would improve performance in any case, we identify several interesting phenomena. (i) We identify a *diminishing return* phenomenon: if the consensus distance stays below a critical value (critical consensus distance), decreasing the consensus distance further does not yield any additional performance

[*]Equal contribution [1]EPFL, Lausanne, Switzerland. Correspondence to: Tao Lin <tao.lin@epfl.ch>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021.

gains. For the main interests of this work, deep learning training, we (ii) identify the pivotal initial training phase where the critical consensus distance matters and the training consensus distance heavily influences the final training and generalization performance, and (iii) large consensus distance in later training phases can even be beneficial.

Our findings have far-reaching consequences for practice: By (iv) using consensus control as a principled tool to find, adaptively during training, the appropriate trade-off between targeted generalization performance and affordable communication resources, it is possible to exploit the efficiency benefits of decentralized methods without sacrificing quality. While our numerical study, on Computer Vision (CV) tasks (CIFAR-10 and ImageNet-32) as well as Natural Language Processing (NLP) tasks (transformer models for machine translation), mainly focuses on the data-center setting with homogeneous nodes, our findings also apply to decentralized training over time-varying topologies and the more difficult heterogeneous setting alike.

2. Related Work

2.1. Decentralized Learning

For general decentralized optimization, common algorithms are either gradient-based methods with gossip averaging steps (Kempe et al., 2003; Xiao & Boyd, 2004; Boyd et al., 2006), or problem-structure dependent methods, such as primal-dual methods (Hong et al., 2017; Sun & Hong, 2019). In this work, we focus on non-convex decentralized deep learning problems and only consider gradient-based methods with gossip averaging—methods that do not support stochastic gradients (not suitable for deep learning) are omitted for the discussion.

The convergence rate of gossip averaging towards the consensus among the nodes can be expressed in terms of the (expected) spectral gap of the mixing matrix. Lian et al. (2017) combine SGD with gossip averaging for deep learning and show that the leading term in the convergence rate $\mathcal{O}\big(\frac{1}{n\varepsilon^2}\big)$ is consistent with the convergence of the centralized mini-batch SGD (Dekel et al., 2012) and the spectral gap only affects the asymptotically smaller terms. Similar results have been observed very recently for related schemes (Scaman et al., 2017; 2018; Koloskova et al., 2019; 2020a;b; Vogels et al., 2020). To reduce the communication overhead (number of peer-to-peer communications), sparse topologies have been studied recently (Assran et al., 2019; Wang et al., 2019; 2020a; Nadiradze et al., 2020). Whilst a few recent works focus on the impact of the topology on the optimization performance (Luo et al., 2019; Neglia et al., 2020), we here identify the consensus distance as a more canonical parameter that can characterize the overall effect of decentralized learning, beyond only the topology. Through this, we are able to provide deeper understanding of the more fine-grained impact of the evolution of the actual consensus distance on the optimization/generalization performance of deep learning.

Prior works propose to either perform a constant number of gossip steps every round (Tsianos & Rabbat, 2016; Scaman et al., 2017; Jiang et al., 2017; 2018; Sharma et al., 2019) to increase the averaging quality, or choose carefully tuned learning rates (Tsitsiklis, 1984; Nedić & Ozdaglar, 2009; Duchi et al., 2012; Yuan et al., 2016) to improve the convergence. However, these works do not analyze the varying effect of consensus distance in the phases of training explicitly. In contrast, we identify the existence of *critical* consensus distance, *adapt* gossip step numbers to the target distance on the fly, and provide insights into how consensus distance at different training phases impacts the decentralized deep learning.

Appendix B.1 further details the relationship between consensus distance and other training metrics influential to the final performance (e.g. gradient diversity in Yin et al. (2018); Johnson et al. (2020)). Besides, we connect the insights into better generalization (Lin et al., 2020b) with other interpretations in Izmailov et al. (2018); Gupta et al. (2020).

2.2. Critical Learning Phase in Deep Learning

The connection between optimization and generalization of deep learning training is not fully understood. A line of work on understanding the early phase of training has recently emerged as a promising avenue for studying this connection. For instance, Keskar et al. (2017); Sagun et al. (2018); Achille et al. (2019); Golatkar et al. (2019); Frankle et al. (2020) point out the existence of a "critical phase" for regularizing deep networks, which is decisive for the final generalization ability. Achille et al. (2019); Jastrzebski et al. (2019); Fort & Ganguli (2019); Jastrzebski et al. (2020) empirically demonstrate the rapid change in the local shape of the loss surface in the initial training phase.

In this work, we reach a similar conclusion for decentralized deep learning: we identify the importance of the initial training phase through the lens of consensus distance.

3. Theoretical Understanding

In this section, we study the trade-off between training performance and the exactness of parameter averaging, and establish the notion of critical consensus distance.

For the sake of simplicity, we consider decentralized stochastic gradient descent (D-SGD) without momentum in this section, and focus on the optimization difficulty in our theoretical analysis. Theoretically analyzing the generalization performance for deep learning is an open problem and not intended in this work. Instead we provide extensive empirical evaluation, addressing generalization for both D-SGD with and without momentum in Section 4.

All proofs are deferred to Appendix C.

3.1. Notation and Setting

The agents are tasked to solve a sum-structured optimization problem $f\colon \mathbb{R}^d \to \mathbb{R}$ of the form

$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^d} \left[f(\mathbf{x}) := \tfrac{1}{n}\sum_{i=1}^n f_i(\mathbf{x}) \right] , \qquad (1)$$

where the components $f_i : \mathbb{R}^d \to \mathbb{R}$ are distributed among the n nodes and are given in stochastic form: $f_i(\mathbf{x}) := \mathbb{E}_{\xi\sim\mathcal{D}_i}\left[F_i(\mathbf{x},\xi)\right]$, where $\mathcal{D}_i$ denotes the local data distribution on node $i \in [n]$. For data-center settings, where data is re-shuffled periodically among nodes, these distributions are identical, but in other scenarios there can be differences between nodes. In D-SGD, each agent $i \in [n]$ maintains local parameters $\mathbf{x}_i^{(t)} \in \mathbb{R}^d$, and updates them as:

$$\mathbf{x}_i^{(t+1)} = \sum_{j=1}^n w_{ij}\left(\mathbf{x}_j^{(t)} - \eta\nabla F_j(\mathbf{x}_j^{(t)}, \xi_j^{(t)})\right) , \qquad \text{(D-SGD)}$$

that is, by a stochastic gradient step based on a sample $\xi_i^{(t)} \sim \mathcal{D}_i$, followed by gossip averaging with neighboring nodes in the network encoded by the mixing weights w_{ij}. As parameters can differ across nodes, we define $\bar{\mathbf{x}} := \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$ and $\mathbf{X} := [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, and $\bar{\mathbf{X}} := [\bar{\mathbf{x}}, \dots, \bar{\mathbf{x}}] \equiv \mathbf{X}\frac{1}{n}\mathbf{1}\mathbf{1}^\top$.

Assumption 1 (Mixing matrix)**.** *Every sample of the (possibly randomized) mixing matrix $\mathbf{W} = \{w_{ij}\} \in \mathbb{R}^{n\times n}$ is doubly stochastic and there exists a parameter $p > 0$ s.t.*

$$\mathbb{E}_{\mathbf{W}}\left\|\mathbf{X}\mathbf{W} - \bar{\mathbf{X}}\right\|_F^2 \le (1-p)\left\|\mathbf{X} - \bar{\mathbf{X}}\right\|_F^2, \forall \mathbf{X} \in \mathbb{R}^{d\times n}. \quad (2)$$

This assumption covers a broad variety of settings (see e.g. Koloskova et al., 2020b), such as D-SGD with fixed (constant) mixing matrix with spectral gap ρ, with parameter $p = 1 - (1-\rho)^2 = \Theta(\rho)$, but also for randomly chosen mixing matrices, for instance random matchings.

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

Assumption 3 (Bounded noise σ and diversity ζ)**.** *There exists constants σ^2, ζ^2 s.t. $\forall \mathbf{x}_1, \dots \mathbf{x}_n \in \mathbb{R}^d$*

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\xi_i}\|\nabla F_i(\mathbf{x}_i,\xi_i) - \nabla f_i(\mathbf{x}_i)\|_2^2 &\le \sigma^2 , \\ \frac{1}{n}\sum_{i=1}^n \|\nabla f_i(\mathbf{x}_i) - \nabla f(\mathbf{x}_i)\|_2^2 &\le \zeta^2 . \end{aligned} \qquad (3)$$

3.2. Decentralized Consensus Optimization

Under the above standard assumptions in decentralized optimization, the convergence rate of (D-SGD) has been shown as follows:

Theorem 3.1 (Koloskova et al. (2020b))**.** *Let f_i be L-smooth and stepsize $\gamma \le \gamma_{\max} = \mathcal{O}\big(\frac{p}{L}\big)$. Then there exists an optimal stepsize $\gamma \le \gamma_{\max}$ such that $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t)})\right\|_2^2 \le \varepsilon$ for*

$$T = \mathcal{O}\left(\frac{\sigma^2}{n\varepsilon^2} + \frac{\sqrt{p}\sigma + \zeta}{p\varepsilon^{3/2}} + \frac{1}{p\varepsilon}\right)\cdot L(f(\mathbf{x}_0) - f^\star) .$$

In comparison, for centralized mini-batch SGD (C-SGD) we are allowed to choose a potentially much larger stepsize $\gamma'_{\max} = \mathcal{O}\big(\frac{1}{L}\big)$, and can bound the number of iterations by $\mathcal{O}\big(\frac{\sigma^2}{n\varepsilon^2} + \frac{1}{\varepsilon}\big)$. While asymptotically both these rates are equivalent, they differ in the low accuracy setting when ε is not too small. That is, especially in the first phase of optimization where the lower order terms matter.

As our first theoretical contribution, we show that if the individual iterates of the agents stay sufficiently close, then D-SGD can converge as fast as C-SGD. To measure this difference between agents, we use the *consensus distance*

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

Proposition 3.2 (Critical Consensus Distance (CCD))**.** *If the consensus distance is bounded by*

$$\Xi_t^2 \le \left(\frac{1}{Ln}\gamma\sigma^2 + \frac{1}{8L^2}\left\|\nabla f(\bar{\mathbf{x}}^{(t)})\right\|^2 =: \Gamma_t^2\right) \qquad (4)$$

for all t, then in D-SGD we may choose larger stepsizes $\gamma \le \gamma'_{\max} = \mathcal{O}\big(\frac{1}{L}\big)$ and recover the convergence rate of C-SGD, that is $\mathcal{O}\big(\frac{\sigma^2}{n\varepsilon^2} + \frac{1}{\varepsilon}\big)$ (Dekel et al., 2012; Bottou et al., 2018). We refer to Γ_t^2 as critical consensus distance *(CCD).*

Note that the CCD does not depend on the graph topology and that $\Gamma_t^2 > 0$, which means that we do not need perfect consensus between agents to recover the C-SGD rate, but we allow consensus distance $\Xi_t^2 \ge 0$ (i.e. the $\Xi_t^2 = 0\ \forall t$, as we have for centralized optimization, is sufficient but not necessary). In Section 4, we empirically examine the existence of the critical consensus distance Ξ_t^2 in decentralized deep learning, as we cannot compute the critical consensus distance in a closed-form (through L and σ^2).

We now estimate the magnitude of the consensus distance in D-SGD and compare it to the CCD.

Proposition 3.3 (Typical consensus distance)**.** *Let $\phi_t^2 := \frac{1}{n}\sum_{i=1}^n \left\|\nabla f_i(\mathbf{x}_i^{(t)})\right\|^2$. Then under the assumption that γ, p are constant, and the ϕ_t does not change too fast between iterations, i.e. not decreasing faster than exponentially: $\phi_t^2 \le (1 + p/4)\phi_{t+1}^2$, the consensus distance in D-SGD satisfies*

$$\Xi_t^2 = (1-p)\gamma^2\cdot\mathcal{O}\left(\frac{\phi_t^2}{p^2} + \frac{\sigma^2}{p}\right) . \qquad (5)$$

While these assumptions do not hold in epochs with learning rate decay, we observe in practice that during epochs of a constant learning rate the gradients indeed do not change too fast (see Figure 6(b)). Thus these assumptions are reasonable approximations to capture the practical behavior.

3.3. Controlling the Consensus Distance

We now investigate scenarios where the typical consensus distance derived in Proposition 3.3 *can* be smaller than the critical value (CCD). This reveals two orthogonal strategies to control the consensus distance in D-SGD. We here assume diversity $\zeta = 0$ as with i.i.d. training data, and that the stepsize $\gamma \leq \mathcal{O}\big(\frac{1}{L}\big)$ as for C-SGD, and give a more refined discussion in Appendix C.3.

Learning rate decay (changing γ). We observe that when $\gamma = \mathcal{O}\big(\frac{p}{nL}\big)$ then $\Xi_t^2 \leq \Gamma_t^2$ (if the noise σ is small, especially for $\sigma = 0$, then the weaker assumption $\gamma = \mathcal{O}\big(\frac{p}{L}\big)$ is sufficient). However, choosing too small stepsizes can impact performance in practice. In C-SGD the constraint on the stepsize is loose ($\gamma \leq \frac{1}{L}$). Yet, after sufficient learning rate decay, the desired CCD can be reached.

More gossip iterations (changing p). We observe that when $\frac{1}{1-p} = \mathcal{O}(1 + \gamma L n)$, then $\Xi_t^2 \leq \Gamma_t^2$ (again, when the noise σ is small, especially when $\sigma^2 = 0$, a weaker condition $\frac{1}{1-p} = \mathcal{O}(1+\gamma L)$ is sufficient). Whilst designing new mixing topologies to control p might not be possible due to practical constraints (fixed network, denser graphs increase latency, etc.), a simple and commonly used strategy is to use repeated gossip steps in every round.

Lemma 3.4 (Repeated gossip (Xiao & Boyd, 2004; Boyd et al., 2006))**.** *Suppose* $\mathbf{W} = \mathbf{W}_k \ldots \mathbf{W}_1$*, for* k *(possibly randomized) mixing matrices with parameter* p *each. Then the mixing parameter for* $\mathbf{W}$ *is at least* $p_{\mathbf{W}} \geq 1 - (1-p)^k$.

From this, we see that the mixing parameter can be improved exponentially when applying more gossip steps. To ensure $p_{\mathbf{W}} \geq 1 - \frac{1}{1+\gamma L n}$, at most $k \leq \frac{\ln(1+\gamma L n)}{p} = \tilde{\mathcal{O}}\big(\frac{1}{p}\big)$ repetitions are required.

4. Inspecting Consensus Distance for Decentralized Training

Our analysis in Section 3 shows that we can—at least in theory—recover the convergence behavior of C-SGD by controlling the consensus distance. Now, we direct our focus on generalization in decentralized deep learning training. We show, empirically (not theoretically, see also Appendix B.2), that the critical consensus distance is an important metric to capture the connection between optimization and generalization in deep learning—e.g. Figure 2 in Section 4.3 showcases that by addressing the optimization difficulty in the critical initial training phase (Figure 2(a) and Figure 2(b)), the final generalization gap can be perfectly closed (Figure 2(c), Table 2 and Table 3).

First we introduce and justify our experimental design in Section 4.1. We describe the implementation in Section 4.2. In Section 4.3, we present our findings on image classification benchmark with standard SGD optimizer, which is the main focus of this work; a preliminary study on Transformer with Adam optimizer and inverse square root learning rate schedule can be found in Section 4.4.

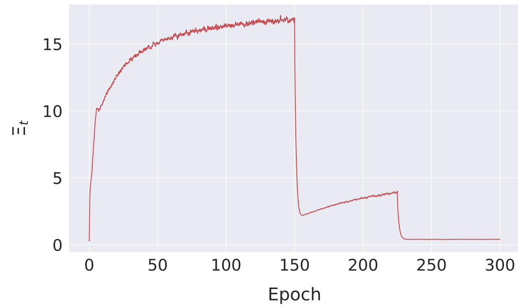


Figure 1: **Evolution of the consensus distance** Ξ for ResNet-20 on CIFAR-10 ($n\!=\!32$) with ring topology.

4.1. Experiment Design: Controlled Training Phases

Phase-wise training. Since the consensus distance evolves throughout training, identifying its impact at every training step is infeasible. However, as the consensus distance and critical consensus distance (CCD) both significantly depend on the learning rate (Propositions 3.2 and 3.3), we expect rather consistent observations during phases in which the learning rate is kept fixed and more drastic changes between such phases. On CV tasks, stage-wise learning rate schedule is the common practice for SOTA distributed training as described in Section 4.2: thus the training can be naturally divided into phases through the learning rate decay[1], in each of which training dynamics are significantly different from the others, such as Ξ_t (Figure 1), ϕ_t (Figure 6(b)) and L-smoothness (Figure 6(c)). The transformer (NLP task) has no well-defined training phases due to the conventional inverse square root learning rate, thus for the sake of simplicity, we consider the entire transformer training as one phase as a preliminary study.

Individual phase investigation. In order to eliminate the coupling of effects from other phases, in each experiment we place only one phase under consensus distance control (the control refers to perform multiple gossip steps as in Section 3.3 to reach certain distance targets), while performing exact averaging (All-Reduce for all nodes) on model parameters for the other unstudied phases. We demonstrate in Table 5 of Section 4.3 that the decentralization impacts on different phases are rather orthogonal, which justifies our design of examining one phase at a time.
For the ease of presentation, the term "phase-x" refers to a training phase between $(x\!-\!1)$-th and x-th learning rate decay. The notation "dec-phase-x" indicates that only in "phase-x" the model is trained with a decentralized com-

[1] The learning rate warmup is only over a very small fraction of training epochs (e.g. 5 out of 300 epochs on CIFAR-10). To simplify the analysis, we do not consider it as a separate phase.

munication topology, while for other phases we perform All-Reduce on model parameters. We compare the result of each individually decentralized phase with that of All-Reduce centralized training (on all training phases), so as to identify when (which phase) and how decentralized training influences final generalization performance.

4.2. Experimental Setup

Datasets and models. We empirically study the decentralized training behavior on the following two tasks, on convolutional neural networks and transformer architectures: (1) Image Classification for CIFAR-10 (Krizhevsky & Hinton, 2009) and ImageNet-32 (i.e. image resolution of 32) (Chrabaszcz et al., 2017), with the standard data augmentation and preprocessing scheme (He et al., 2016); and (2) Neural Machine Translation for the Multi30k dataset (Elliott et al., 2016). For Image Classification, ResNet-20 (He et al., 2016) with different widths are used on CIFAR (default width of 1) and ImageNet-32 (width factor of 3)[2]. For Neural Machine Translation, a down-scaled transformer architecture (by 2 w.r.t. the base model in Vaswani et al. (2017)) is used. Weight initialization schemes follow Goyal et al. (2017); He et al. (2015) and Vaswani et al. (2017) respectively. Unless mentioned otherwise, our experiments are repeated over three random seeds.

Training schemes. We use mini-batch SGD with a Nesterov momentum of 0.9 without dampening for image classification task (we confirm our findings in Section 4.3 for SGD without momentum), and Adam is used for neural machine translation task. Unless mentioned otherwise we use the optimal learning rate (lr) from centralized training for our decentralized experiments[3] in order to observe the impact of *decentralization* on normal *centralized* training.

- For image classification experiments, unless mentioned otherwise, the models are trained for 300 and 90 epochs for CIFAR-10 and ImageNet-32 respectively; the local mini-batch size are set to 32 and 64. By default, all experiments follow the SOTA learning rate scheme in distributed deep learning literatures (Goyal et al., 2017; He et al., 2019) with learning rate scaling and warmup scheme. The learning rate is always gradually warmed up from a relatively small value (i.e. 0.1) for the first 5 epochs. Besides, the learning rate will be divided by 10 when the model has accessed specified fractions of the total number of training samples (He et al., 2016); we use $\{\frac{1}{2}, \frac{3}{4}\}$ and $\{\frac{1}{3}, \frac{2}{3}, \frac{8}{9}\}$ for CIFAR and ImageNet respectively. All results in tables are test top-1 accuracy.
- For experiments on neural machine translation, we use standard inverse square root learning rate schedule (Vaswani et al., 2017) with local mini-batch size 64. The warm-up step is set to 4000 for the mini-batch size of 64 and is linearly scaled down by the global mini-batch size.

Consensus distance control. For consensus control, we adopt the "more gossip iterations" strategy introduced in Section 3.3. That is, we perform multiple gossip steps (if needed) until reaching the desired target consensus distance value. Two metrics are considered to set the consensus distance target value during the specified training phase:

- constant target distance (main approach[4]): the target consensus distance Ξ for a phase is the *maximum consensus distance* $\Xi_{\max}$ of the *current phase* in normal (uncontrolled) decentralized training, multiplied by a factor. For a given topology, the smaller the factor, the tighter the consensus.
- adaptive target distance (in Appendix E.3.1): the target consensus distance Ξ for the current step is the averaged local gradient norm ϕ_t^{avg} scaled by a factor. For stability, we use the exponentially moving averaged value ϕ_t^{ema} of ϕ_t^{avg} (accumulated during the corresponding phase).

We use a ring as the main decentralized communication topology, as it is a particularly hard instance with a small spectral gap (cf. Table 10) which allows us to study a wide range of target consensus distances by modifying the number of gossip steps (in appendix we show consistent findings on time varying exponential topology in Table 18 and 19)..

4.3. Findings on Computer Vision Tasks

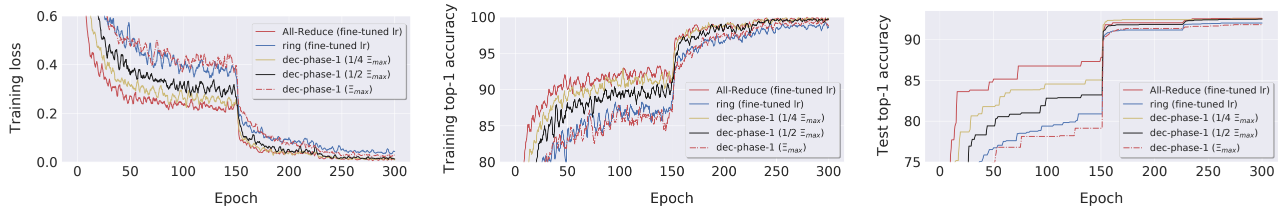
In this section we present our empirical findings and provide insights into how consensus distance at different phases impacts the training generalization for CV tasks (i.e. CIFAR-10, Imagenet-32).

Critical consensus distance exists in the initial training phase—consensus distance below this critical threshold ensures good optimization and generalization. In the initial training phase, both training and generalization performance are heavily impacted by the consensus distance ("dec-phase-1" in Figure 2 and Table 2). A smaller consensus distance in the early phase results in considerably faster optimization (training loss) and higher generalization performance (test accuracy), and these advantages persist

[2] It takes $\sim$ 7h to finish 1 round of standard ImageNet-32 training with $n = 16$ V100 on a ring, and the cost increases to e.g. 12h for our consensus distance controlled experiments. It is infeasible to perform sufficient experiments on datasets of larger scales with our computation budget.

[3] We find that fine-tuning the learning rate for decentralized experiments does not change our conclusions. E.g., no significant difference can be found for the curves at phase-1 for "ring (fine-tuned lr)" and "dec-phase-1 ($\Xi_{\max}$)" in Figure 2(a) and 2(b). We have similar observations in Table 14 after the sufficient learning rate tuning on phase-1.

[4] We use this one primarily since we can directly regulate the magnitude of consensus distance. In experiments, target $\Xi = \Xi_{\max}$ refers to the normal (i.e. uncontrolled) decentralized training.



(a) Training loss. (b) Training top-1 accuracy. (c) Test top-1 accuracy.

Figure 2: Learning curves for ResNet-20 on CIFAR-10 ($n = 32$). We compare fine-tuned normal (w/o control) decentralized training (i.e. "ring") with dec-phase-1 on different target consensus distances.

Table 2: **The impact of consensus distance of different phases on generalization performance** (test top-1 accuracy) of training ResNet-20 on CIFAR-10 on ring. The All-Reduce performance for $n = 32$ and $n = 64$ are 92.82 ± 0.27 and 92.71 ± 0.11 respectively. The fine-tuned normal (w/o control) decentralized training performance for $n = 32$ and $n = 64$ are 91.74 ± 0.15 and 89.87 ± 0.12 respectively.

target Ξ / # nodes	dec-phase-1			dec-phase-2			dec-phase-3		
	Ξ_{max}	1/2 Ξ_{max}	1/4 Ξ_{max}	Ξ_{max}	1/2 Ξ_{max}	1/4 Ξ_{max}	Ξ_{max}	1/2 Ξ_{max}	1/4 Ξ_{max}
n=32	91.78 ± 0.35	92.36 ± 0.21	92.74 ± 0.10	93.04 ± 0.01	92.99 ± 0.30	92.87 ± 0.11	92.60 ± 0.00	92.82 ± 0.21	92.85 ± 0.24
n=64	90.31 ± 0.12	92.18 ± 0.07	92.45 ± 0.17	93.14 ± 0.04	92.94 ± 0.10	92.79 ± 0.07	92.23 ± 0.12	92.50 ± 0.09	92.60 ± 0.10

Table 3: **The impact of different consensus distances on generalization for different phases** of training ResNet-20-3 on ImageNet-32 on ring. The centralized baseline performances for $n = 16$ and $n = 32$ are 51.74 ± 0.06 and 51.98 ± 0.37 respectively, while those of decentralized training (on a fixed ring) are 51.04 ± 0.06 and 50.17 ± 0.04. The reported test top-1 accuracies are over two seeds.

target Ξ / # nodes	dec-phase-1			dec-phase-2			dec-phase-3			dec-phase-4		
	Ξ_{max}	1/2 Ξ_{max}	1/4 Ξ_{max}	Ξ_{max}	1/2 Ξ_{max}	1/4 Ξ_{max}	Ξ_{max}	1/2 Ξ_{max}	1/4 Ξ_{max}	Ξ_{max}	1/2 Ξ_{max}	1/4 Ξ_{max}
n=16	51.22 ± 0.08	51.79 ± 0.10	51.71 ± 0.03	51.59 ± 0.02	51.67 ± 0.01	51.65 ± 0.13	51.80 ± 0.10	51.81 ± 0.13	51.81 ± 0.04	51.72 ± 0.02	51.76 ± 0.01	51.74 ± 0.06
n=32	50.76 ± 0.18	51.27 ± 0.07	51.60 ± 0.21	51.39 ± 0.07	51.59 ± 0.04	51.66 ± 0.12	51.79 ± 0.06	51.73 ± 0.10	51.77 ± 0.10	51.70 ± 0.02	51.71 ± 0.02	51.70 ± 0.02

over the entire training.

When the consensus distance is larger (e.g. 1/2 Ξ_{max} for CIFAR-10), the optimization (training performance) can eventually catch up with the centralized convergence (c.f. Figure 2(a) and 2(b)) but a considerable generalization gap still remains (92.36 v.s. 92.82 for the setup in Figure 2) as shown in Table 2. A consistent pattern can be found in ImageNet-32 experiments[5], as shown in Table 3. These observations to some extent are consistent with the insights of the critical learning phase described in Golatkar et al. (2019); Jastrzebski et al. (2020); Frankle et al. (2020) for centralized training, where it is argued that the initial learning phase is crucial for the final generalization.

Notably, perfect consensus distance is not required to recover the centralized training performance. For instance, 1/4 Ξ_{max} is sufficient in CIFAR-10 experiments to approach the optimal centralized training performance in both optimization and *generalization* at the end. Smaller distances (e.g. 1/8 Ξ_{max}, 1/16 Ξ_{max}) do not bring significant gain (92.77 and 92.72 respectively in Table 12). The performance saturates (c.f. 92.74 for 1/4 Ξ_{max}) with significantly increased communication overhead (e.g. Figure 10 of Appendix E.1). This confirms that our analysis and discovery in Section 3 are sensible in the initial training phase: *there exists a critical consensus distance for the training, below which the impact of decentralization is negligible.*

A non-negligible consensus distance at middle phases can improve generalization over centralized training. Surprisingly, it is not always the case that the generalization performance improves with a shrinking consensus distance. We observe that at the phase right after the initial training plateaus (e.g. phase-2 for CIFAR-10, phase-3 for Imagenet-32), a non-negligible consensus distance[6] actually boosts the generalization performance over the centralized training which has been deemed optimal. In CIFAR-10 dec-phase-2 experiments (Table 2), the generalization performance increases monotonically with the evaluated consensus distance and is consistently superior to that of the centralized training (e.g. 93.04, 92.99, 92.87 over 92.82 for $n = 32$). Analogous observation can be obtained in Imagenet-32 dec-phase-3 experiments (Table 3).

This coincides with the observations firstly introduced in post-local SGD (Lin et al., 2020b), where for better generalization, consensus distance is created among local machines by less frequent model parameter synchronization (All-Reduce) in late training phases (e.g. phase-2, phase-3 for CIFAR). Thus non-negligible consensus distance at middle phases can be viewed as a means of injecting proper

[5] 1/2 Ξ_{max} has already been tight enough to recover the centralized performance for ImageNet-32 ($n = 32$), while a significant performance drop can be observed between Ξ_{max} and 1/2 Ξ_{max}.

[6] Table 19 of Appendix E.3.1 shows that there exists optimal consensus distance at middle phases, beyond which the gain in generalization (brought by noise injection) starts to diminish.

Table 4: **The impact of consensus distance on generalization performance with vanilla SGD (without momentum)** (test top-1 accuracy) of training ResNet-20 on CIFAR-10 on ring. The All-Reduce performance for $n = 32$ and $n = 64$ are 90.64 ± 0.19 and 90.58 ± 0.26 respectively. The fine-tuned normal (w/o control) decentralized training performance for $n=32$ and $n=64$ are 90.30 ± 0.14 and 88.92 ± 0.23 respectively. We repeat experiments for $n=32$ for 3 seeds and $n=64$ for 2 seeds.

target Ξ / # nodes	dec-phase-1			dec-phase-2		
	$\Xi_{\max}$	$1/2\Xi_{\max}$	$1/4\Xi_{\max}$	$\Xi_{\max}$	$1/2\Xi_{\max}$	$1/4\Xi_{\max}$
$n = 32$	90.51 ± 0.05	90.74 ± 0.14	90.88 ± 0.37	90.64 ± 0.18	90.55 ± 0.19	90.57 ± 0.17
$n = 64$	88.80 ± 0.03	89.89 ± 0.03	90.43 ± 0.05	90.63 ± 0.37	90.46 ± 0.15	90.63 ± 0.25

noise as argued in Lin et al. (2020b), which reduces communication cost and in the meanwhile benefits generalization.

At the last phase of training, the consensus distance only marginally impacts the generalization performance. Similar to the initial training phase, the final convergence phase seems to favor small consensus distances in CIFAR-10 experiments. However, its impact is less prominent in comparison: for dec-phase-3, performance of a smaller consensus distance (1/4 $\Xi_{\max}$) is only 0.25% and 0.37% higher than that of $\Xi_{\max}$ for $n = 32, 64$ respectively (Table 2). In Imagenet-32 experiments, dec-phase-3 performance is not even affected by changes in consensus.

Quality propagation across phases. Our previous experiments only consider a single phase of decentralized training. We now evaluate the lasting impact of consensus across the sequence of multiple phases. In Table 5, we control the consensus distance for both phase-1 and phase-2 when training on CIFAR-10. Our previous findings hold when we view each controlled phase separately. For instance, when we apply 1/2 $\Xi_{\max}$ consensus control to phase-2 (the middle column in Table 5), we can still observe that a smaller consensus distance in phase-1 results in a higher performance as in our previous finding. Hence our previous findings are valid in more general cases of decentralized training.

Longer training cannot close the generalization gap caused by large consensus distances in the initial training phase. As discussed above, large consensus distances in the initial phase can result in significant generalization loss. Table 6 investigates whether a prolonged training on the initial phase can address this difficulty: we prolong the phase-1 for CIFAR-10 with a range of consensus distances

Table 5: **Quality propagation across training phases with different consensus distances** on ResNet-20 for CIFAR-10 (Ring with $n=32$). In phase-1 and phase-2, the model parameters reach inexact consensus of different target consensus distance Ξ, while phase-3 performs All-Reduce on model parameters.

phase-1 / phase-2	$\Xi_{\max}$	1/2 $\Xi_{\max}$	1/4 $\Xi_{\max}$
1/2 $\Xi_{\max}$	92.48 ± 0.19	92.46 ± 0.11	92.31 ± 0.23
1/4 $\Xi_{\max}$	92.73 ± 0.11	92.66 ± 0.08	92.69 ± 0.19
1/8 $\Xi_{\max}$	93.10 ± 0.22	92.88 ± 0.15	92.91 ± 0.06

Table 6: **The impact of different numbers of training epochs (at phase-1)** on generalization, for training ResNet-20 on CIFAR-10 (dec-phase-1 with $n=32$). The number of epochs at phase-1 is chosen from $\{150, 200, 250\}$, while the other training setting is identical to that of dec-phase-1 in Table 2.

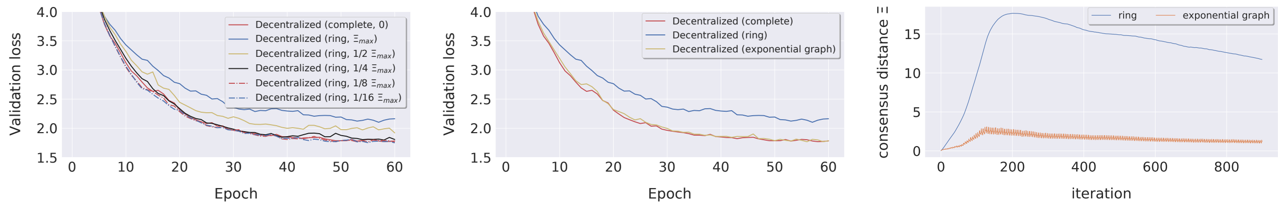
target Ξ	training epochs at phase-1		
	150	200	250
$\Xi_{\max}$	91.78 ± 0.35	91.91 ± 0.19	92.04 ± 0.14
1/2 $\Xi_{\max}$	92.36 ± 0.21	92.55 ± 0.07	92.67 ± 0.13
1/4 $\Xi_{\max}$	92.74 ± 0.10	92.91 ± 0.15	92.84 ± 0.20

and leave the other training phases centralized. We can observe that although longer training is beneficial for each consensus distance, it cannot recover the generalization gap resulting from large consensus distance. For instance, the maximum gain (among all evaluated cases) of increasing the epoch number from 150 to 250 is 0.31% at 1/2 $\Xi_{\max}$, which is lower than the average gain (around 0.6%) of merely reducing the consensus distance from $\Xi_{\max}$ to 1/2 $\Xi_{\max}$. Table 15 in Appendix E.2 evaluates cases where dec-phase-2 and dec-phase-3 are prolonged. We find longer training in these two phases brings about negligible performance gain.

Consistent findings on decentralized SGD without momentum. To validate the coherence between our theory and experiments, we perform similar consensus distance control experiments on vanilla SGD optimizer (i.e. without momentum) for dec-phase-1 and dec-phase-2 on CIFAR-10. The patterns illustrated in Table 4 are consistent with our previous observations in Table 2 and Table 3, supporting the claim on the relation between consensus distance and generalization performance (which stands regardless of the use of momentum).

4.4. Preliminary study on training transformer models

The critical consensus distance also exists in NLP tasks. Figure 3(a) demonstrates that 1/4 $\Xi_{\max}$ target control on a ring is sufficient to recover the centralized training performance. Besides, the target consensus distance in this case can be reached by exponential graph (and thus target test performance, as shown in Figure 3(b) and 3(c)). These justify the importance of designing an efficient communication topology/scheme in practice so as to effectively reach the CCD.



(a) Different target Ξs. (b) Decentralized baseline. (c) Consensus distance Ξ.

Figure 3: Learning curves for training Transformer on Multi30k ($n = 32$).

Table 7: **The importance of phase-1** for training ResNet-20 on CIFAR-10 ($n = 32$), in terms of **(1) target consensus distance** and **(2) the number of training epochs**. In phase-2 and phase-3, we perform *decentralized* training (w/o consensus distance control).

# of epochs \ target Ξ	Ξ_{max}	1/2 Ξ_{max}	1/4 Ξ_{max}	1/8 Ξ_{max}	0 Ξ_{max}
150	91.74 ± 0.15	92.31 ± 0.12	92.81 ± 0.22	92.91 ± 0.15	92.94 ± 0.07
200	91.81 ± 0.22	92.88 ± 0.20	93.00 ± 0.18	93.01 ± 0.10	92.90 ± 0.17
250	92.09 ± 0.23	92.74 ± 0.11	93.15 ± 0.26	92.99 ± 0.24	93.31 ± 0.06

5. Impact on Practice

Practical guidelines: prioritizing the initial training phase. Apart from effectiveness (generalization/test performance), efficiency (time) stands as the other crucial goal in deep learning, and thus how to allocate communication resource over the training becomes a relevant question.

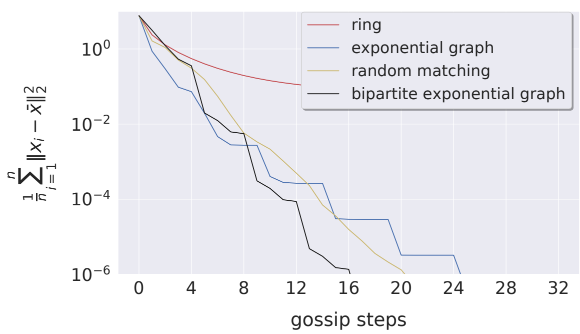


Figure 4: **Consensus distance evolution against the number of gossip steps** on different topologies ($n = 32$). The initial x_i's are sampled uniformly from $[0, 10]$. Results on different topology scales are deferred to Appendix E.1.

As indicated by our first empirical finding (and theory in Section 3), the initial training phase bears the greatest importance over all other training phases; therefore the communication expenditure should be concentrated on the initial phase to maintain a consensus distance lower than the CCD. We suggest a list of communication topologies with superior spectral properties, i.e. exponential graph (Assran et al., 2019) and random matching (Nadiradze et al., 2020) in Figure 4 (the definition of the topology is detailed in Appendix E.1), which be utilized to achieve fast convergence in gossip averaging.

The late training phases should be less prioritized for communication resources, due to the generalization benefits from a reasonable consensus distance in the middle phases. Providing a rigorous way to quantify the optimal consensus distance is non-trivial, and is left as future work.

In Table 7 we show that the above-mentioned guideline is practically feasible: as long as the quality of the initial phase is ensured, we can afford to slacken the consensus control for later phases, in particular the middle phase. For instance, when the number of epochs is 150, a consensus control of 1/4 Ξ_{max} in the initial phase with uncontrolled middle and final phase is adequate to recover the centralized training performance (92.81 v.s. 92.82). Note that here the noise injection from the uncontrolled middle phase also contributes positively to the performance. Table 18 in Appendix E.3.1 additionally justifies the applicability of applying this guideline on exponential graphs.

Practical implementation of Consensus Control in Data-Centers. Computing the exact consensus distance requires the average of all model parameters in $\mathbb{R}^d$, which is prohibitively expensive (All-Reduce). We propose therefore to use the following efficient quantity estimator

$$\Theta_t^2 := \frac{1}{n}\sum_{i=1}^{n}\theta_i^{(t)} \quad \text{with} \quad \theta_i^{(t)} := \left\| \sum_{j=1}^{n} w_{ij}\mathbf{x}_j^{(t)} - \mathbf{x}_i^{(t)} \right\|_2^2 ,$$

instead (in Lemma A.1 we prove that $\Xi_t \leq \frac{2}{p}\Theta_t$ is an upper-bound of consensus distance and thus a valid control parameter, see also Section A.2 for numerical validation). The values $\theta_i^{(t)} \in \mathbb{R}$ can be computed *locally* on each node when updating the parameters at negligible cost (compared to gradient computations), and computing Θ_t requires only averaging of scalars. While this can be implemented efficiently in data-centers (the cost of averaging these scalar values is negligible compared to averaging high-dimensional parameter vectors in the gossip steps), this might not be efficient over arbitrary decentralized network.

Table 8 and 9 in Appendix A.2 show the feasibility of integrating the control of Θ_t with our practical guidelines for efficient training in data-centers, which serves as a strong starting point for designing decentralized training algorithms with a desired balance between communication cost and training performance.

6. Conclusion

In this work, we theoretically identify the consensus distance as an essential factor for decentralized training. We show the existence of a critical consensus distance, below which the consensus distance does not hinder optimization. Our deep learning experiments validate our theoretical findings and extend them to the generalization performance. Based on these insights, we propose practical guidelines for favorable generalization performance with low communication expenses, on arbitrary communication networks.

While we focused in this work on data-center training with iid data as an important first step, consensus control may be of even greater importance in non-iid scenarios (such as in Hsieh et al., 2020).

Acknowledgements

We acknowledge funding from a Google Focused Research Award, Facebook, and European Horizon 2020 FET Proactive Project DIGIPREDICT.

References

Achille, A., Rovere, M., and Soatto, S. Critical learning periods in deep networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BkeStsCcKQ.

Assran, M., Loizou, N., Ballas, N., and Rabbat, M. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, pp. 344–353. PMLR, 2019.

Bellet, A., Guerraoui, R., Taziki, M., and Tommasi, M. Personalized and private peer-to-peer machine learning. In *AISTATS - Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 473–481. PMLR, 09–11 Apr 2018.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. Randomized gossip algorithms. *IEEE transactions on information theory*, 52 (6):2508–2530, 2006.

Chrabaszcz, P., Loshchilov, I., and Hutter, F. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

Dekel, O., Gilad-Bachrach, R., Shamir, O., and Xiao, L. Optimal distributed online prediction using mini-batches. *The Journal of Machine Learning Research*, 13:165–202, 2012.

Duchi, J. C., Agarwal, A., and Wainwright, M. J. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, 57(3):592–606, 2012. doi: 10.1109/TAC.2011.2161027.

Elliott, D., Frank, S., Sima'an, K., and Specia, L. Multi30k: Multilingual english-german image descriptions. *arXiv preprint arXiv:1605.00459*, 2016.

Fort, S. and Ganguli, S. Emergent properties of the local geometry of neural loss landscapes. *arXiv preprint arXiv:1910.05929*, 2019.

Frankle, J., Schwab, D. J., and Morcos, A. S. The early phase of neural network training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hkl1iRNFwS.

Golatkar, A. S., Achille, A., and Soatto, S. Time matters in regularizing deep networks: Weight decay and data augmentation affect early learning dynamics, matter little near convergence. In *Advances in Neural Information Processing Systems*, pp. 10678–10688, 2019.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Gupta, V., Serrano, S. A., and DeCoste, D. Stochastic weight averaging in parallel: Large-batch training that generalizes well. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygFWAEFwS.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., and Li, M. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 558–567, 2019.

Hong, M., Hajinezhad, D., and Zhao, M.-M. Prox-pda: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks. In *International Conference on Machine Learning*, pp. 1529–1538, 2017.

Hsieh, K., Phanishayee, A., Mutlu, O., and Gibbons, P. The non-IID data quagmire of decentralized machine learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 4387–4398. PMLR, 2020.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Jastrzebski, S., Kenton, Z., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. On the relation between the sharpest directions of DNN loss and the SGD step length. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkgEaj05t7.

Jastrzebski, S., Szymczak, M., Fort, S., Arpit, D., Tabor, J., Cho*, K., and Geras*, K. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1g87C4KwB.

Jiang, Z., Balu, A., Hegde, C., and Sarkar, S. Collaborative deep learning in fixed topology networks. In *NIPS - Advances in Neural Information Processing Systems*, volume 30, 2017.

Jiang, Z., Balu, A., Hegde, C., and Sarkar, S. On consensus-optimality trade-offs in collaborative deep learning. *arXiv preprint arXiv:1805.12120*, 2018.

Johnson, T., Agrawal, P., Gu, H., and Guestrin, C. Adascale sgd: A user-friendly algorithm for distributed training. In *International Conference on Machine Learning*, pp. 4911–4920. PMLR, 2020.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Kempe, D., Dobra, A., and Gehrke, J. Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pp. 482–491. IEEE, 2003.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017.

Koloskova, A., Stich, S. U., and Jaggi, M. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *ICML 2019 - Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 3479–3487. PMLR, 2019. URL http://proceedings.mlr.press/v97/koloskova19a.html.

Koloskova, A., Lin, T., Stich, S. U., and Jaggi, M. Decentralized deep learning with arbitrary communication compression. In *ICLR - International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=SkgGCkrKvH.

Koloskova, A., Loizou, N., Boreiri, S., Jaggi, M., and Stich, S. U. A unified theory of decentralized SGD with changing topology and local updates. In *International Conference on Machine Learning*, 2020b.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2009.

Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu, J. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *NIPS - Advances in Neural Information Processing Systems*, pp. 5330–5340, 2017.

Lian, X., Zhang, W., Zhang, C., and Liu, J. Asynchronous decentralized parallel stochastic gradient descent. In *International Conference on Machine Learning*, pp. 3043–3052. PMLR, 2018.

Lin, T., Kong, L., Stich, S., and Jaggi, M. Extrapolation for large-batch training in deep learning. In *ICML - International Conference on Machine Learning*, 2020a.

Lin, T., Stich, S. U., Patel, K. K., and Jaggi, M. Don't use large mini-batches, use local SGD. In *ICLR - International Conference on Learning Representations*, 2020b. URL https://openreview.net/forum?id=B1eyO1BFPr.

Luo, Q., Lin, J., Zhuo, Y., and Qian, X. Hop: Heterogeneity-aware decentralized training. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 893–907, 2019.

Nadiradze, G., Sabour, A., Alistarh, D., Sharma, A., Markov, I., and Aksenov, V. Decentralized sgd with asynchronous, local and quantized updates. *arXiv preprint arXiv:1910.12308*, 2020.

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

Neglia, G., Xu, C., Towsley, D., and Calbi, G. Decentralized gradient methods: does topology matter? In *AISTATS*, 2020.

Neyshabur, B. Implicit regularization in deep learning. *PhD Thesis*, abs/1709.01953, 2017.

Sagun, L., Evci, U., Guney, V. U., Dauphin, Y., and Bottou, L. Empirical analysis of the hessian of over-parametrized neural networks. *ICLR workshop*, 2018.

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. How does batch normalization help optimization? In *NeurIPS - Advances in Neural Information Processing Systems*, pp. 2483–2493, 2018.

Scaman, K., Bach, F., Bubeck, S., Lee, Y. T., and Massoulié, L. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *International Conference on Machine Learning*, 2017.

Scaman, K., Bach, F., Bubeck, S., Massoulié, L., and Lee, Y. T. Optimal algorithms for non-smooth distributed optimization in networks. In *NeurIPS - Advances in Neural Information Processing Systems*, pp. 2740–2749, 2018.

Shallue, C. J., Lee, J., Antognini, J., Sohl-Dickstein, J., Frostig, R., and Dahl, G. E. Measuring the effects of data parallelism on neural network training. *arXiv preprint arXiv:1811.03600*, 2018.

Sharma, C., Narayanan, V., and Balamurugan, P. A simple and fast distributed accelerated gradient method. In *OPT2019: 11th Annual Workshop on Optimization for Machine Learning*, 2019.

Stich, S. U. and Karimireddy, S. P. The error-feedback framework: Better rates for sgd with delayed gradients and compressed updates. *Journal of Machine Learning Research*, 21:1–36, 2020.

Sun, H. and Hong, M. Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms. *IEEE Transactions on Signal processing*, 67(22):5912–5928, 2019.

Tsianos, K. I. and Rabbat, M. G. Efficient distributed online prediction and stochastic optimization with approximate distributed averaging. *IEEE Transactions on Signal and Information Processing over Networks*, 2(4):489–506, 2016.

Tsitsiklis, J. N. *Problems in decentralized decision making and computation*. PhD thesis, Massachusetts Institute of Technology, 1984.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Vogels, T., Karimireddy, S. P., and Jaggi, M. Powergossip: Practical low-rank communication compression in decentralized deep learning. In *NeurIPS 2020 - Thirty-fourth Conference on Neural Information Processing Systems*, 2020.

Wang, J., Sahu, A. K., Yang, Z., Joshi, G., and Kar, S. Matcha: Speeding up decentralized sgd via matching decomposition sampling. In *2019 Sixth Indian Control Conference (ICC)*, pp. 299–300, 2019. doi: 10.1109/ICC47138.2019.9123209.

Wang, J., Sahu, A. K., Joshi, G., and Kar, S. Exploring the error-runtime trade-off in decentralized optimization. In *2020 54th Asilomar Conference on Signals, Systems, and Computers*, pp. 910–914, 2020a. doi: 10.1109/IEEECONF51394.2020.9443529.

Wang, J., Tantia, V., Ballas, N., and Rabbat, M. SlowMo: Improving Communication-Efficient Distributed SGD with Slow Momentum. In *International Conference on Learning Representations*, 2020b. URL https://openreview.net/forum?id=SkxJ8REYPH.

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

Yin, D., Pananjady, A., Lam, M., Papailiopoulos, D., Ramchandran, K., and Bartlett, P. Gradient diversity: a key ingredient for scalable distributed learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1998–2007. PMLR, 2018.

You, Y., Zhang, Z., Hsieh, C.-J., Demmel, J., and Keutzer, K. Imagenet training in minutes. In *Proceedings of the 47th International Conference on Parallel Processing*, pp. 1–10, 2018.

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