SADDLe: Sharpness-Aware Decentralized Deep Learning with Heterogeneous Data

Sakshi Choudhary
Purdue University
West Lafayette, IN
choudh23@purdue.edu

Sai Aparna Aketi
Purdue University
West Lafayette, IN
aketiaparna@gmail.com

Kaushik Roy
Purdue University
West Lafayette, IN
kaushik@purdue.edu

Abstract

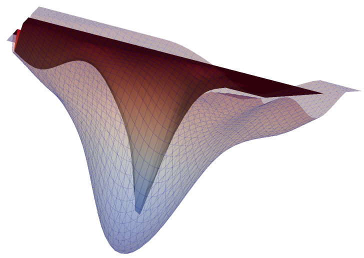
Decentralized training enables learning with distributed datasets generated at different locations without relying on a central server. In realistic scenarios, the data distribution across these sparsely connected learning agents can be significantly heterogeneous, leading to local model over-fitting and poor global model generalization. Another challenge is the high communication cost of training models in such a peer-to-peer fashion without any central coordination. In this paper, we jointly tackle these two-fold practical challenges by proposing SADDLe, a set of sharpness-aware decentralized deep learning algorithms. SADDLe leverages Sharpness-Aware Minimization (SAM) to seek a flatter loss landscape during training, resulting in better model generalization as well as enhanced robustness to communication compression. We present two versions of our approach and demonstrate its effectiveness through extensive experiments on various Computer Vision datasets (CIFAR-10, CIFAR-100, Imagenette, and ImageNet), model architectures, and graph topologies. Our results show that SADDLe leads to 1-20% improvement in test accuracy as compared to existing techniques while incurring a minimal accuracy drop ($\sim 1\%$*) in the presence of up to 4*$\times$* compression.*

1. Introduction

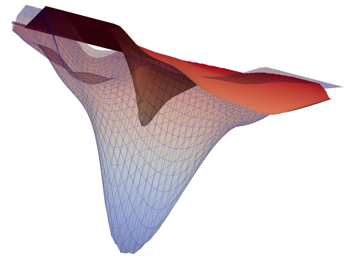
Federated learning enables training with distributed data across multiple agents under the orchestration of a central server [29]. However, the presence of such a central entity can lead to a single point of failure and network bandwidth issues [6]. To address these concerns, several decentralized learning algorithms have been proposed [1, 2, 6, 13, 26, 33, 34]. Decentralized learning is a peer-to-peer learning paradigm in which agents connected in a fixed graph topology learn by communicating with their peers/neighbors without the need for a central server. Decentralized learning algorithms have been shown to perform comparably to centralized algorithms on image classification and natural language processing (NLP) tasks [33]. The authors in [33] present Decentralized Parallel Stochastic Gradient Descent (DPSGD), which combines SGD with gossip averaging algorithm [48] and show that the convergence rate of DPSGD is similar to its centralized counterpart [10]. Decentralized Momentum Stochastic Gradient Descent [7] introduced momentum to DPSGD, while Stochastic Gradient Push (SGP) [6] extends DPSGD to directed and time-varying graphs.

The above-mentioned algorithms assume the data to be independently and identically distributed (IID) across the agents. This refers to a scenario where the training data is distributed uniformly and randomly. However, in real-world applications, the data distributions can be remarkably different, i.e. non-IID or heterogeneous [17]. While several algorithms have been proposed to mitigate the impact of such data heterogeneity [1, 2, 13, 26, 34, 46], these algorithms do not explicitly focus on the aspect of communication cost, which may account for about 70% of energy consumption [16, 38]. In decentralized learning, agents communicate the models with their neighbors after every mini-batch update, leading to high communication costs. Various communication compression techniques have been proposed to address this, but these algorithms primarily focus on the settings when the data distribution is IID [27, 45, 47].

In this paper, we aim to answer the following question: *Can we improve the performance of decentralized learning on heterogeneous data in terms of test accuracy as well as robustness to communication compression?* We put forward an orthogonal direction of enhancing the local training at each agent to positively impact the global model generalization. We propose that seeking a flatter loss landscape during training can alleviate the issue of local over-fitting, a common concern in decentralized learning scenarios with non-IID data. To achieve this, we propose SADDLe, a set of sharpness-aware decentralized deep learning algorithms. SADDLe improves generalization by simultaneously min-



(a) QGM vs Q-SADDLe



(b) Comp QGM vs Comp Q-SADDLe

Figure 1. Loss landscape visualization for QGM (surface) vs Q-SADDLe (mesh) and Comp QGM (surface) vs Comp Q-SADDLe (mesh) for ResNet-20 trained on CIFAR-10 with non-IID data across 10 agents. Comp signifies communication compression through 8-bit stochastic quantization.

imizing the loss value and the sharpness through gradient perturbation. This is enabled by utilizing Sharpness-Aware Minimization (SAM) [14] to seek parameters in neighborhoods with uniformly low loss values. Furthermore, flatter loss landscapes are inherently more robust to perturbations in the training loss landscape [14]. Leveraging this potential, we observe that SAM leads to enhanced robustness against compression errors due to erroneous model updates resulting from communication compression. Notably, training with SAM optimizer at each agent also reduces compression errors, a result we attribute to SAM producing lower gradient norms, which serve as an upper bound on compression errors (as discussed in Section 4.2).

To that effect, we demonstrate that SADDLe can be used in synergy with existing decentralized learning algorithms for non-IID data to attain better generalization and reduce the accuracy drop incurred due to compression. Specifically, we present two versions of our approach: Q-SADDLe, which incorporates a Quasi Global Momentum (QGM) buffer [34], and N-SADDLe, which utilizes cross-gradient information [2]. Figure 1 presents a visualization of the loss landscape [32] for QGM, Q-SADDLe, and their compressed counterparts. Clearly, Q-SADDLe has a much smoother loss landscape, resulting in better generalization and minimal performance loss due to communication compression. Our detailed theoretical analysis highlights that the convergence rate of Q-SADDLe matches the well-known best result in decentralized learning [33]. We also conduct extensive experiments to establish that Q-SADDLe and N-SADDLe achieve better accuracy than state-of-the-art decentralized algorithms [2,34], with a minimal accuracy drop due to communication compression.

In summary, we make the following contributions:

- We propose Sharpness-Aware Decentralized Deep Learning (SADDLe) to seek flatter loss landscapes in decentralized learning, alleviating the local over-fitting with non-IID data.
- Leveraging the fact that flatter loss landscapes tend to be more robust to perturbations, we demonstrate that SADDLe improves robustness to communication compression in the presence of data heterogeneity.
- We theoretically establish that SADDLe leads to a convergence rate of $\mathcal{O}(1/\sqrt{nT})$, similar to existing decentralized learning algorithms [33].
- Through extensive experiments on various datasets, models, graphs, and compression schemes, we show that Q-SADDLe and N-SADDLe result in a 1-20% improvement in test accuracy. Additionally, our proposed algorithms maintain a minimal accuracy drop of 1% for up to 4× compression, in contrast to the 4.3% average accuracy drop for the baselines.

2. Related Work

Data Heterogeneity. The impact of data heterogeneity in decentralized learning is an active area of research [1, 2, 13, 26, 34, 46]. Quasi-Global Momentum (QGM) [34] improves decentralized learning with non-IID data through a globally synchronized momentum buffer. Gradient Tracking [26] tracks average gradients but requires 2x communication overhead as compared to DPSGD [33], while Global Update Tracking [1] tracks the average model updates to enhance performance with heterogeneous data. Cross Gradient Aggregation (CGA) [13] and Neighborhood Gradient Mean (NGM) [2] utilize cross-gradient information through an extra communication round, achieving state-of-the-art performance in terms of test accuracy. In this work, we take an orthogonal route and focus on improving local training with a flatness-seeking optimizer [14] to achieve better generalization.

Communication Compression. Several algorithms have been proposed for communication-restricted decentralized settings [27, 44, 45, 49]. DeepSqueeze [45] introduced error-compensated compression to decentralized learning. Choco-SGD [27] communicates compressed model updates rather than parameters and achieves better accuracy than DeepSqueeze. Recently, BEER [49] adopted communication compression with gradient tracking [26], resulting in a faster convergence rate than Choco-SGD [27]. However, as shown in QGM [34], gradient tracking doesn't scale well for deep learning models and requires further study. In this paper, we compress the first (and only) communication round in Q-SADDLe and the second round in N-SADDLe. In both cases, we observe that SADDLe aids communication efficiency by alleviating the severe accuracy degradation incurred due to compression in existing decentralized learning algorithms for non-IID data [2, 34].

Sharpness-Aware Minimization. Sharpness-Aware Minimization (SAM) [14] explores the connection between the flatness of minima and generalization by simultaneously minimizing loss value and loss sharpness during training [23, 25]. The authors in [5] provide a theoretical understanding of SAM through convergence results. Several variants of SAM have been proposed for centralized learning [12, 22, 31, 36, 37, 50]. In addition, there have been several efforts to improve the generalization performance in federated learning using SAM [8, 9, 39, 40, 43]. The authors in [51] provide some theoretical insights to establish an asymptotic equivalence between decentralized training and average-direction SAM. In contrast, our work focuses on simultaneously improving test accuracy and robustness to communication compression for decentralized learning with extreme data heterogeneity.

3. Background

A global model is learned in decentralized learning by aggregating models trained on locally stored data at n agents connected in a sparse graph topology. This topology is modeled as a graph $G = ([n], \mathbf{W})$, where $\mathbf{W}$ is the mixing matrix indicating the graph's connectivity. Each entry w_{ij} in $\mathbf{W}$ encodes the effect of agent j on agent i, and $w_{ij} = 0$ implies that agents i and j are not connected directly. $\mathcal{N}(i)$ represents neighbors of i including itself. We aim to minimize the global loss function $f(\mathbf{x})$ shown in equation 1. Here, $F_i(\mathbf{x}; d_i)$ is the local loss function at agent i, and $f_i(\mathbf{x})$ is the expected value of $F_i(\mathbf{x}; d_i)$ over the dataset D_i.

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x});\ f_i(\mathbf{x}) = \mathbb{E}_{d_i\sim D_i}[F_i(\mathbf{x}; d_i)] \tag{1}$$

DPSGD [33] tackles this by combining Stochastic Gradient Descent (SGD) with gossip averaging algorithms [48]. Each agent maintains model parameters $\mathbf{x}_i^t$, computes local gradient $\mathbf{g}_i^t$ through SGD, and incorporates neighborhood information as shown in the following update rule:

$$\text{DPSGD: } \mathbf{x}_i^{t+1} = \sum_{j\in\mathcal{N}(i)} w_{ij}\mathbf{x}_j^t - \eta\mathbf{g}_j^t; \mathbf{g}_j^t = \nabla F_j(\mathbf{x}_j^t, d_j^t).$$

DPSGD assumes the data distribution across the agents to be IID and results in significant performance degradation in the presence of data heterogeneity. To handle this, QGM [34] incorporates a globally synchronized momentum buffer within DPSGD. This mitigates the impact of non-IID data by maintaining a form of global information through the momentum buffer, resulting in better test accuracy without any extra communication overhead. To further improve the performance with extreme heterogeneity, NGM [2] and CGA [13] utilize cross-gradients obtained through an additional communication round. In the first communication round, the agents exchange models with each other (similar to DPSGD). However, in the second round, the agents communicate cross-gradients computed over the neighbors' models and their local data. Each gradient update is a weighted average of the self and received cross-gradients [2]. Note that these algorithms represent two distinct variants proposed to enhance decentralized learning with non-IID data based on the available communication budget. For additional details, refer to Supplementary Section 2.1.

4. Methodology

This section presents the two variants of SADDLe and their communication-compressed versions.

4.1. SADDLe

In the presence of data heterogeneity, models in decentralized training tend to overfit the local data at each agent. Aggregating such models adversely impacts the global model's generalization ability. To circumvent this, we propose SADDLe, which purposely seeks a flatter loss landscape in each training iteration through Sharpness-Aware Minimization (SAM) [14]. Instead of focusing on finding parameters with low loss values like SGD, SAM searches for parameters whose neighborhoods have uniformly low loss. This is achieved by adding a perturbation ξ_i to the model parameters, which is obtained through a scaled gradient ascent step. To summarize, SADDLe aims to solve the following optimization problem:

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{d_i\sim D_i} \max_{\|\xi_i\|\le\rho} [F_i(\mathbf{x}_i^t + \xi_i; d_i)] \ \ \forall i, \quad \text{where } \ \xi_i = \rho\frac{\mathbf{g}_i}{\|\mathbf{g}_i\|} \tag{2}$$

Here, ρ is a tunable hyperparameter, defining the perturbation radius. Since SADDLe modifies the local optimizer at each agent, it is orthogonal to existing techniques and can be used in synergy to improve performance with non-IID data. We employ a QGM buffer [34] and cross-gradients similar to NGM [2] with SADDLe and present two versions: Q-SADDLe and N-SADDLe.

Algorithm 1 QGM v.s. Q-SADDLe

Input: Each agent $i \in [1, n]$ initializes model parameters $\mathbf{x}_i$, $\hat{\mathbf{m}}_i^{(0)} = 0$, step size η, momentum coefficients β, μ, mixing matrix $\mathbf{W} = [w_{ij}]_{i,j \in [1,n]}$, $\mathcal{N}(i)$ represents neighbors of i including itself.

procedure TRAIN( ) for $\forall i$
1. **for** t = $1, 2, \ldots, T$ **do**
2. $\quad \mathbf{g}_i^t = \nabla F_i(\mathbf{x}_i^t; d_i^t)$ for $d_i^t \sim D_i$
3. $\quad \mathbf{m}_i^{(t)} = \beta \hat{\mathbf{m}}_i^{(t-1)} + \mathbf{g}_i^{(t)}$
4. $\quad \tilde{\mathbf{g}}_i^t = \nabla F_i(\mathbf{x}_i^t + \xi(\mathbf{x}_i^t); d_i^t)$, where $\xi(\mathbf{x}_i^t) = \rho \frac{\mathbf{g}_i^t}{\|\mathbf{g}_i^t\|}$
5. $\quad \mathbf{m}_i^{(t)} = \beta \hat{\mathbf{m}}_i^{(t-1)} + \tilde{\mathbf{g}}_i^{(t)}$
6. $\quad \mathbf{x}_i^{(t+1/2)} = \mathbf{x}_i^{(t)} - \eta \mathbf{m}_i^{(t)}$
7. $\quad$ SENDRECEIVE($\mathbf{x}_i^{(t+1/2)}$)
8. $\quad \mathbf{x}_i^{t+1} = \sum_{j \in \mathcal{N}_i^{(t)}} w_{ij} \mathbf{x}_j^{(t+1/2)}$
9. $\quad \mathbf{d}_i^{(t)} = \frac{\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)}}{\eta}$
10. $\quad \hat{\mathbf{m}}^{(t)} = \mu \hat{\mathbf{m}}_i^{(t-1)} + (1-\mu) \mathbf{d}_i^{(t)}$
11. **end**

return $\mathbf{x}_i^T$

The differences between QGM and Q-SADDLe are highlighted in Algorithm 1. In particular, Q-SADDLe utilizes the SAM-based gradient update $\tilde{\mathbf{g}}_i$ shown in line 4 instead of the gradient update $\mathbf{g}_i$. N-SADDLe employs SAM-based self and cross-gradients to further improve the performance of NGM. Algorithm 1 in the Supplementary material summarizes the difference in training procedures for NGM and N-SADDLe.

4.2. SADDLe with Compressed Communication

A major concern in decentralized learning is the high communication cost of training. Hence, we also investigate the impact of a flatter loss landscape on generalization performance in the presence of communication compression. We present compressed versions of QGM and Q-SADDLe in Algorithm 2. Instead of sharing models $\mathbf{x}_i$, the agents exchange compressed model updates $\mathbf{q}_i$ (similar to Choco-SGD [27]). Each agent maintains compressed copies $\hat{\mathbf{x}}_j$ of their neighbors and employs a modified gossip averaging step as shown on line 7 (Algorithm 2). Similarly, we implement Comp NGM and Comp N-SADDLe to compress the second communication round, which involves sharing cross-gradients (Algorithm 2 in Supplementary). In addition to robustness to data heterogeneity, seeking flatter models also results in higher resiliency to compression error (as indicated by our results in Tables 1-6).

Algorithm 2 Comp QGM v.s. Comp Q-SADDLe

Input: Each agent $i \in [1, n]$ initializes model parameters $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i^1 = 0$, step size η, momentum coefficients β, μ, global averaging rate γ, mixing matrix $\mathbf{W} = [w_{ij}]_{i,j \in [1,n]}$.

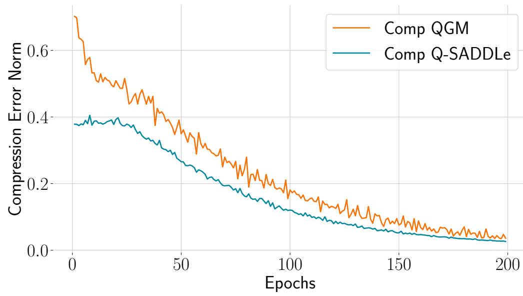
procedure TRAIN( ) for $\forall i$
1. **for** t = $1, 2, \ldots, T$ **do**
2. $\quad \mathbf{g}_i^t = \nabla F_i(\mathbf{x}_i^t; d_i^t)$ for $d_i^t \sim D_i$
3. $\quad \mathbf{m}_i^{(t)} = \beta \hat{\mathbf{m}}_i^{(t-1)} + \mathbf{g}_i^{(t)}$
4. $\quad \tilde{\mathbf{g}}_i^t = \nabla F_i(\mathbf{x}_i^t + \xi(\mathbf{x}_i^t); d_i^t)$, where $\xi(\mathbf{x}_i^t) = \rho \frac{\mathbf{g}_i^t}{\|\mathbf{g}_i^t\|}$
5. $\quad \mathbf{m}_i^{(t)} = \beta \hat{\mathbf{m}}_i^{(t-1)} + \tilde{\mathbf{g}}_i^{(t)}$
6. $\quad \mathbf{x}_i^{(t+1/2)} = \mathbf{x}_i^{(t)} - \eta \mathbf{m}_i^{(t)}$
7. $\quad \mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t+1/2)} + \gamma \sum_{j \in \mathcal{N}(i)} w_{ij} (\hat{\mathbf{x}}_j^{(t)} - \hat{\mathbf{x}}_i^{(t)})$
8. $\quad \mathbf{d}_i^{(t)} = \frac{\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)}}{\eta}$
9. $\quad \hat{\mathbf{m}}^{(t)} = \mu \hat{\mathbf{m}}_i^{(t-1)} + (1-\mu) \mathbf{d}_i^{(t)}$
10. $\quad \mathbf{q}_i^{(t)} = Q(\mathbf{x}_i^{(t+1)} - \hat{\mathbf{x}}_i^{(t)})$
11. $\quad$ SENDRECEIVE($\mathbf{q}_i^{(t)}$)
12. $\quad \hat{\mathbf{x}}_j^{(t+1)} = \mathbf{q}_j^{(t)} + \hat{\mathbf{x}}_j^{(t)}$ for all $j \in \mathcal{N}(i)$
13. **end**

return $\mathbf{x}_i^T$

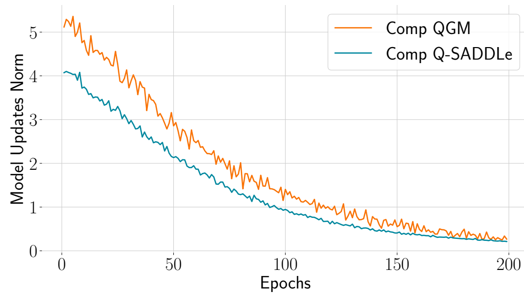
Interestingly, Comp Q-SADDLe and Comp N-SADDLe incur less compression error than Comp QGM and Comp NGM respectively, leading to a lower accuracy drop due to compression. We investigate this with the aid of a well-known bound on the compression error [3]. For a compression operator $Q(.)$, the expectation of error $\|Q(\theta) - \theta\|$ is bounded as:

$$\mathbb{E}_Q \|Q(\theta) - \theta\|^2 \leq (1-\zeta)\|\theta\|^2, \text{ where } \zeta > 0 \quad (3)$$

In our setup, θ corresponds to model updates $(\mathbf{x}_i - \hat{\mathbf{x}}_i)$ for Comp QGM and Comp Q-SADDLe, and gradients for Comp NGM and Comp N-SADDLe. Note that a wide range of compression operators (with some ζ) have been shown to adhere to this bound [3, 4, 27, 28, 42]. Figure 2 shows the norm of compression error (i.e. $\|Q(\theta) - \theta\|$) and the norm of model updates (i.e. $\|\theta\| = \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|$) for Comp QGM and Comp Q-SADDLe for ResNet-20 trained on CIFAR-10 in a 10 agent ring with extreme data heterogeneity. It can be observed that Q-SADDLe leads to a lower norm of model updates ($\|\theta\|$) and lower compression error. In essence, the bound in Equation 3 is tighter for Q-SADDLe than QGM in the presence of compression. We observe similar trends for N-SADDLe (refer to Figure 1 in Supplementary). Note that



(a) Compression Error ($||Q(\theta) - \theta||$)



(b) Model Updates ($||\mathbf{x}_i - \hat{\mathbf{x}}_i||$)

Figure 2. Impact of flatness on (a) Compression Error and (b) Model Updates for ResNet-20 trained on CIFAR-10 distributed in a non-IID manner across a 10 agent ring topology.

our observation regarding lower gradient norms (and hence model updates) in the presence of SAM aligns with the fact that SAM optimization is a special form of penalizing the gradient norm [50].

5. Convergence Rate Analysis

In this section, we provide a convergence analysis for Q-SADDLe. Similar to prior works in decentralized learning [33, 34], we make the following standard assumptions:

Assumption 1 - Lipschitz Gradients: Each function $f_i(\mathbf{x})$ is L-smooth i.e., $||\nabla f_i(\mathbf{y}) - \nabla f_i(\mathbf{x})|| \leq L||\mathbf{y} - \mathbf{x}||$.

Assumption 2 - Bounded Variance: The variance of the stochastic gradients is assumed to be bounded. There exist constants σ and δ such that

$$\begin{aligned} &\mathbb{E}_{d\sim\mathcal{D}_i}||\nabla F_i(\mathbf{x};d) - \nabla f_i(\mathbf{x})||^2 \leq \sigma^2 \\ &\frac{1}{n}\sum_{i=1}^{n}||\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})||^2 \leq \delta^2 \quad \forall i \end{aligned} \tag{4}$$

Assumption 3 - Doubly Stochastic Mixing Matrix: $\mathbf{W}$ is a real doubly stochastic matrix which satisfies $\mathbb{E}_{\mathbf{W}}\left\|\mathbf{ZW} - \bar{\mathbf{Z}}\right\|^2 \leq (1-\lambda)\left\|\mathbf{Z} - \bar{\mathbf{Z}}\right\|^2$ for any matrix $\mathbf{Z} \in \mathbb{R}^{d\times n}$ and $\bar{\mathbf{Z}} = \mathbf{Z}\frac{1}{n}^T$.

Theorem 1 presents convergence of the proposed Q-SADDLe algorithm (proof in Supplementary Section 1.1).

Theorem 1 *Given Assumptions 1-3, for a momentum coefficients β and μ, let the learning rate satisfy $\eta \leq \min\left(\frac{\lambda}{7L}, \frac{1-\beta}{4L}, \frac{(1-\beta)^2(1-\mu)}{\sqrt{12}L\beta}\right)$. For all $T \geq 1$, we have*

$$\begin{aligned} \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[||\nabla f(\bar{\mathbf{x}}_t)||^2\right] &\leq \frac{4}{\tilde{\eta}T}(\mathbb{E}[f(\bar{x}^0) - f^*]) + \left(\frac{6\tilde{\eta}L}{n} + \right. \\ &\left. 18\tilde{\eta}^2L^2C_2 + \frac{768\tilde{\eta}^2L^2C_1}{\lambda^2}\right)\sigma^2 + \left(\frac{832L^2\tilde{\eta}^2(1-\beta)^2}{\lambda^2}\right)\delta^2 \\ &+ \left(8L^2 + \frac{12\tilde{\eta}L^3}{n} + 36\tilde{\eta}^2L^4C_2 + \frac{3136L^4\tilde{\eta}^2C_1}{\lambda^2}\right)\rho^2 \end{aligned} \tag{5}$$

where $C_1 = \frac{(2-\beta-\mu)(1-\beta)^2}{(1-\mu)}$, $C_2 = \frac{\beta^2}{(1-\mu)(1-\beta)}$, $\bar{x}$ is the average/consensus model and $\tilde{\eta} = \frac{\eta}{(1-\beta)}$.

We observe that the convergence rate includes three main terms related to the suboptimality gap $f(\bar{x}^0) - f^*$, the sampling variance σ and the gradient variance δ representing data heterogeneity, followed by an additional term compared to existing state-of-the-art decentralized convergence bounds [33, 34]. This term includes the perturbation radius ρ, signifying the impact of leveraging gradient perturbation to improve generalization in decentralized learning. We present a corollary to show the convergence rate in terms of training iterations (proof in Supplementary Section 1.2).

Corollary 2 *Suppose that the learning rate satisfies $\eta = \mathcal{O}\left(\sqrt{\frac{n}{T}}\right)$ and $\rho = \mathcal{O}\left(\sqrt{\frac{1}{T}}\right)$. For a sufficiently large T,*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[||\nabla f(\bar{\mathbf{x}}_t)||^2\right] \leq \mathcal{O}\left(\frac{1}{\sqrt{nT}} + \frac{1}{T} + \frac{1}{T^{3/2}} + \frac{1}{T^2}\right) \tag{6}$$

Note that the dominant term here is $(1/\sqrt{nT})$, and the terms introduced because of the additional SGD step for flatness (i.e., $1/T^{3/2}$ and $1/T^2$) can be ignored due to their higher order (similar to [39–41]). This convergence rate matches the well-known best result in existing decentralized learning algorithms [33].

6. Experiments

6.1. Experimental Setup

We analyze the test accuracy and communication efficiency of the proposed Q-SADDLe and N-SADDLe and compare them with the state-of-the-art QGM and NGM. We evaluate the proposed algorithms across diverse datasets, model architectures, graph topologies, graph sizes, and compression operators, with all models using Evonorm [35] as it is better suited for non-IID data [18]. The analysis

Table 1. Test accuracy of QGM, Q-SADDLe, and their compressed versions evaluated on CIFAR-10 and CIFAR-100 over ResNet-20, distributed over ring topologies. Comp implies stochastic quantization [3] with 8 bits, which leads to 4× lower communication cost.

Agents	Comp	Method	CIFAR-10		CIFAR-100	
			$\alpha = 0.01$	$\alpha = 0.001$	$\alpha = 0.01$	$\alpha = 0.001$
5		QGM	88.44 ± 0.39	88.72 ± 0.64	56.84 ± 2.01	59.58 ± 1.22
	✓	QGM	86.85 ± 0.73	86.82 ± 0.99	48.80 ± 8.58	51.99 ± 4.23
		Q-SADDLe (ours)	**90.66 ± 0.08**	**90.56 ± 0.33**	**61.96 ± 1.00**	**61.64 ± 0.70**
	✓	*Q-SADDLe (ours)*	**89.70 ± 0.15**	**90.02 ± 0.08**	**60.11 ± 0.99**	**60.53 ± 0.25**
10		QGM	77.41 ± 8.00	79.48 ± 2.76	48.06 ± 4.36	44.16 ± 6.71
	✓	QGM	76.59 ± 5.95	73.03 ± 4.63	46.14 ± 6.88	43.00 ± 6.55
		Q-SADDLe (ours)	**87.72 ± 1.59**	**86.33 ± 0.24**	**58.06 ± 0.68**	**56.76 ± 0.86**
	✓	*Q-SADDLe (ours)*	**87.82 ± 1.42**	**85.57 ± 1.33**	**57.90 ± 0.82**	**56.27 ± 0.84**
20		QGM	72.20 ± 0.77	62.48 ± 8.56	45.23 ± 3.26	44.48 ± 4.53
	✓	QGM	66.61 ± 6.68	60.30 ± 6.60	43.64 ± 3.68	42.75 ± 1.82
		Q-SADDLe (ours)	**78.41 ± 2.13**	**82.81 ± 0.89**	**52.59 ± 0.48**	**48.20 ± 0.93**
	✓	*Q-SADDLe (ours)*	**80.80 ± 2.20**	**82.18 ± 0.56**	**52.64 ± 1.09**	**48.00 ± 0.85**
40		QGM	70.46 ± 4.14	60.86 ± 0.98	40.15 ± 0.90	38.73 ± 1.47
	✓	QGM	67.81 ± 2.62	57.01 ± 1.88	35.36 ± 1.50	36.04 ± 1.31
		Q-SADDLe (ours)	**77.49 ± 0.83**	**73.54 ± 2.04**	**43.25 ± 1.71**	**41.99 ± 1.27**
	✓	*Q-SADDLe (ours)*	**76.35 ± 0.42**	**72.03 ± 2.12**	**41.75 ± 2.14**	**41.03 ± 0.67**

Table 2. Test accuracy of QGM, Q-SADDLe, and their compressed versions evaluated on Imagenette, distributed over a ring topology. Comp implies stochastic quantization [3] with 10-bits.

Agents	Comp	Method	Imagenette (MobileNet-V2)	
			$\alpha = 0.01$	$\alpha = 0.001$
5		QGM	64.25 ± 11.53	57.67 ± 6.32
	✓	QGM	59.86 ± 17.05	50.14 ± 9.39
		Q-SADDLe	**73.34 ± 0.80**	**72.50 ± 0.21**
	✓	*Q-SADDLe*	**72.64 ± 1.66**	**72.77 ± 0.43**
10		QGM	56.30 ± 4.03	45.82 ± 5.99
	✓	QGM	53.50 ± 5.66	36.71 ± 3.65
		Q-SADDLe	**62.35 ± 3.64**	**63.18 ± 1.59**
	✓	*Q-SADDLe*	**63.35 ± 2.61**	**61.14 ± 0.71**

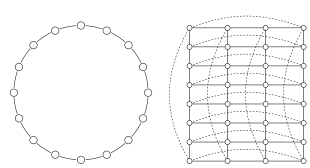


Figure 3. Ring Graph (left), and Torus Graph (right).

is presented on - (a) **Datasets**: CIFAR-10 [30], CIFAR-100 [30], Imagenette [20] and ImageNet [11], (b) **Model architectures**: ResNet-20, ResNet-18 and MobileNet-v2, (c) **Graph topologies**: ring with 2 peers/agent and torus with 4 peers/agent (visualization in Figure 3), (d) **Graph sizes**: 5 to 40 agents, (e) **Compression operators**: Stochastic quantization [3], Top-k sparsification [4, 42] and Sign SGD [24]. For QGM, stochastic quantization diverges beyond 8-10 bits, and Top-k diverges beyond 30% sparsification, likely due to erroneous compressed updates affecting both gossip and momentum buffers (lines 7-8, Algorithm 2). However, in NGM, the second communication round is more compressible as it only affects gradient updates, making 1-bit Sign SGD viable for NGM and N-SADDLe.

We focus on non-IID data partitions generated by Dirichlet distribution [19], varying the concentration parameter α—smaller α increases non-IIDness (see Figure 2 in Supplementary). These partitions are non-overlapping, with no shuffling across the agents during training. Training hyperparameters are detailed in Section 4 of the Supplementary.

6.2. Results

Performance Comparison: As shown in Table 1, for CIFAR-10 Q-SADDLe results in 8.4% better accuracy on average as compared to QGM [34] across a range of graph sizes and two different degrees of non-IIDness ($\alpha = 0.01, 0.001$). QGM suffers a 1-6% accuracy drop in the presence of a stochastic quantization-based compression scheme, whereas, for Q-SADDLe, this drop is only 0-1.5%. For a challenging dataset such as CIFAR-100, Table 1 shows that Q-SADDLe outperforms QGM by ∼6% on average. The accuracy drop due to compression is 1-8% for QGM, while Q-SADDLe proves to be more resilient to compression error with only a 0-1.8% drop in accuracy. We present additional results on Imagenette, a subset of ImageNet trained on MobileNet-v2 in Table 2. Q-SADDLe leads to an average improvement of ∼12% over QGM, with only a 0-2% drop in accuracy due to compression. In contrast, QGM incurs a significant drop of 4-9% in the presence of communication compression. Furthermore, as the degree of non-IIDness is increased from $\alpha = 0.01$ to $\alpha = 0.001$, QGM suffers from an 8.5% average drop in accuracy, whereas Q-SADDLe nearly retains the perfor-

Table 3. Test accuracy of NGM, N-SADDLe, and their compressed versions evaluated on CIFAR-10 and CIFAR-100 over ResNet-20, distributed with different degrees of heterogeneity over ring topologies. Comp implies 1-bit Sign SGD [24] based compression, which reduces the communication cost of the second round by $32\times$ and the total communication cost by $1.94\times$.

Agents	Comp	Method	CIFAR-10		CIFAR-100	
			$\alpha = 0.01$	$\alpha = 0.001$	$\alpha = 0.01$	$\alpha = 0.001$
5		NGM	90.87 ± 0.39	90.73 ± 0.46	59.00 ± 4.26	54.78 ± 4.68
	✓	NGM	89.50 ± 0.68	87.69 ± 1.98	56.91 ± 1.82	50.65 ± 2.67
		N-SADDLe (ours)	**91.96 ± 0.19**	**91.69 ± 0.15**	**63.87 ± 0.45**	**64.10 ± 0.48**
	✓	*N-SADDLe (ours)*	**91.88 ± 0.36**	**91.77 ± 0.19**	**62.35 ± 0.87**	**62.43 ± 0.36**
10		NGM	85.08 ± 2.73	83.43 ± 0.95	55.2 ± 1.41	54.70 ± 1.36
	✓	NGM	76.85 ± 15.10	76.67 ± 3.67	43.41 ± 4.50	43.17 ± 4.65
		N-SADDLe (ours)	**88.43 ± 1.38**	**87.29 ± 1.23**	**59.31 ± 0.61**	**58.37 ± 0.30**
	✓	*N-SADDLe (ours)*	**88.11 ± 1.54**	**87.14 ± 1.45**	**58.39 ± 0.89**	**58.33 ± 0.45**
20		NGM	84.84 ± 0.43	83.58 ± 0.89	53.98 ± 0.31	53.37 ± 0.53
	✓	NGM	83.91 ± 0.96	78.90 ± 0.11	50.07 ± 2.79	46.73 ± 4.35
		N-SADDLe (ours)	**86.26 ± 0.29**	**86.61 ± 0.20**	**55.77 ± 0.53**	**55.14 ± 0.49**
	✓	*N-SADDLe (ours)*	**86.34 ± 0.24**	**87.41 ± 0.52**	**56.65 ± 0.17**	**55.11 ± 1.16**

Table 4. Test accuracy of NGM, N-SADDLe, and their compressed versions evaluated on ImageNette and ImageNet, distributed over a ring topology. Comp implies 1-bit Sign SGD based compression.

Agents	Comp	Method	Imagenette (MobileNet-V2)		ImageNet (ResNet-18)	
			$\alpha = 0.01$	$\alpha = 0.001$	$\alpha = 0.01$	$\alpha = 0.001$
10		NGM	67.68 ± 1.23	67.10 ± 0.71	49.30	46.38
	✓	NGM	63.65 ± 2.70	63.70 ± 0.87	34.66	34.40
		N-SADDLe (ours)	**69.54 ± 0.33**	**68.70 ± 0.79**	**52.20**	**51.94**
	✓	*N-SADDLe (ours)*	**68.09 ± 0.57**	**67.37 ± 0.68**	**51.44**	**49.61**

Table 5. Test accuracy of different decentralized algorithms on CIFAR-10, distributed with $\alpha = 0.001$ over torus topology.

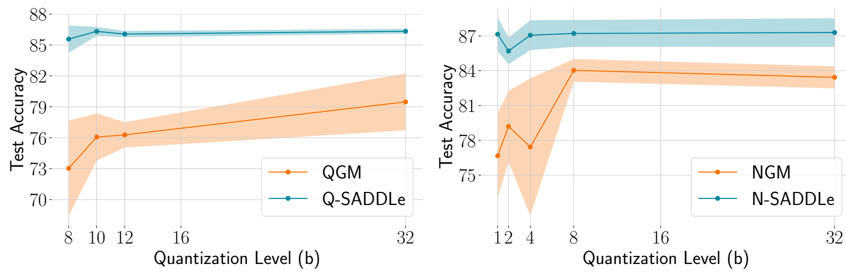
Agents	Comp	Method	Accuracy(%)
40		QGM	57.96 ± 3.90
	✓	QGM	47.08 ± 7.72
		Q-SADDLe (ours)	**70.05 ± 3.35**
	✓	*Q-SADDLe (ours)*	**65.84 ± 2.46**
		NGM	86.00 ± 0.34
	✓	NGM	86.30 ± 0.52
		N-SADDLe (ours)	**86.67 ± 0.32**
	✓	*N-SADDLe (ours)*	**87.00 ± 0.18**

mance. We present additional results for Top-30% Sparsification in Table 2 in Supplementary.

Table 3 and 4 demonstrate the significant improvements in test accuracy and communication efficiency achieved by N-SADDLe over NGM [2]. As shown in Table 3, N-SADDLe outperforms NGM by 2.3% and 4.2% on average for CIFAR-10 and CIFAR-100, respectively. For CIFAR-10, the accuracy drop due to compression for NGM is ∼1-8%, while it is only about 0-0.3% for N-SADDLe. Similarly, for CIFAR-100, the drop due to compression for NGM is ∼2-11%, whereas for N-SADDLe, it is only 0-1.7%. These performance trends are maintained for ImageNette in Table 4, where N-SADDLe outperforms NGM by 1.7%, with a minimal drop of 1.4% with compression (as compared to 3.7% drop in case of NGM). To demonstrate the scalability of our approach, we present additional results on ImageNet distributed over a ring topology of 10 agents with varying degrees of heterogeneity. Our results in Table 4 show that N-SADDLe outperforms NGM by 4.2% while also being much more robust to communication compression. Specifically, NGM incurs a significant drop of 13% in accuracy, compared to about a 1.5% drop for N-SADDLe.

We also evaluate our techniques on a torus graph with 40 agents, and the results are presented in Table 5. Q-SADDLe outperforms QGM by 12%, with only a ∼5% drop in accuracy with compression, whereas QGM experiences a significantly larger drop of ∼11%. N-SADDLe achieves 0.7% better accuracy than NGM, with both methods maintaining their performance even under 1-bit Sign SGD-based communication compression. Additionally, please refer to Table 3 in the Supplementary for results on stochastic quantization-based compression for NGM and N-SADDLe. For the exact communication cost of all our presented experiments, please refer to Section 3.6 in Supplementary.

Impact of Varying Compression Levels: To understand the impact of the degree of compression, we evaluate QGM, Q-SADDLe, NGM, and N-SADDLe for a range of



(a) QGM vs Q-SADDLe (b) NGM vs N-SADDLe

Figure 4. Test accuracy for different levels of quantization-based compression scheme for CIFAR-10 over a 10 agent ring topology.

quantization levels and present the results in Figure 4. Test accuracy for QGM drops from about 79% to 73% as the compression becomes more extreme, while Q-SADDLe retains its performance with a minimal drop of $\sim 0.7\%$. Similarly, NGM incurs an accuracy drop of about $\sim$7% due to compression, while N-SADDLe maintains its performance.

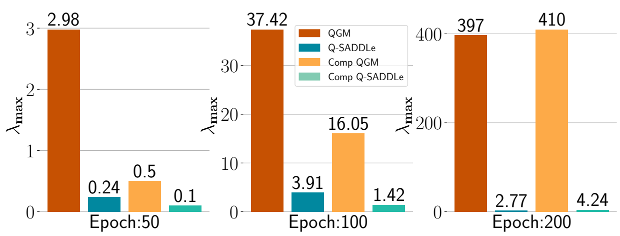


Figure 5. Largest Eigenvalue of the Hessian ($\lambda_{\mathbf{max}}$) at 3 stages of training for ResNet-20 trained on CIFAR-10 in a 10 agent ring topology with α= 0.001.

Evaluating Flatness Measures: To confirm our hypothesis that the presence of SAM in decentralized training leads to a flatter loss landscape, we compute the highest eigenvalue $\lambda_{\mathbf{max}}$ of the Hessian at different epochs during the training [15]. Note that lower the $\lambda_{\mathbf{max}}$, flatter the loss landscape [12, 14, 21]. As shown in Figure 5, Q-SADDLe and Comp Q-SADDLe have consistently lower $\lambda_{\mathbf{max}}$ as compared to QGM and Comp QGM, respectively. This enhances the robustness of Q-SADDLe to erroneous updates due to communication compression. The difference in the eigenvalues is remarkably high towards the end of the training, indicating that models trained with SAM converge to a flatter minimum as expected. Please refer to Figures 3 and 4 in the Supplementary for loss landscape visualization.

Compute-Efficient Variant: SADDLe seeks flatter loss landscapes through a gradient ascent step, requiring an additional backward pass to compute the perturbation ξ_i (Equation 2). To reduce this computational overhead, we implement a more efficient variant of SADDLe, where the gradient ascent step is calculated once in every 5 training iterations [36], leading to 1.66$\times$ lower compute. As shown in Table 6, this compute-efficient Q-SADDLe variant achieves only 1.75% lower accuracy compared to the original version but still outperforms QGM by approximately 10% across two different graph sizes. Even with communication compression, Q-SADDLe nearly maintains its performance, unlike QGM, which incurs a 1-6% drop (Table 1).

Table 6. Test accuracy of compute-efficient *Q-SADDLe* evaluated on CIFAR-10 distributed over ring topology.

Agents	Comp	Accuracy(%)	
		$\alpha = 0.01$	$\alpha = 0.001$
10		85.50 ± 0.26	84.27 ± 0.15
	✓	84.79 ± 0.34	84.61 ± 0.23
20		82.77 ± 1.38	80.07 ± 1.34
	✓	82.41 ± 0.86	80.14 ± 1.67

7. Conclusion

Communication-efficient decentralized learning on heterogeneous data is crucial for enabling on-device learning to leverage vast amounts of user-generated data. In this work, we propose Sharpness-Aware Decentralized Deep Learning (SADDLe) to improve generalization and robustness to communication compression in the presence of data heterogeneity. SADDLe aims to seek a flatter loss landscape during training through gradient perturbation via SAM [14]. Our theoretical analysis shows that SADDLe achieves a convergence rate comparable to well-known decentralized convergence bounds [33]. The proposed technique is complementary to existing decentralized learning algorithms and can be used synergistically to improve performance. We present two versions of our approach, Q-SADDLe, and N-SADDLe, and conduct exhaustive experiments to evaluate these techniques over various datasets, models, graphs, and compression schemes. Our results show that SADDLe leads to 1-20% better accuracy than existing decentralized algorithms for non-IID data, with a minimal drop of $\sim$1% in the presence of up to 4$\times$ communication compression.

Acknowledgements. This work was supported by the Center for the Co-Design of Cognitive Systems (CoCOSYS), an SRC/ DARPA-sponsored JUMP 2.0 center.

References

[1] Sai Aparna Aketi, Abolfazl Hashemi, and Kaushik Roy. Global update tracking: A decentralized learning algorithm for heterogeneous data. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 48939–48961. Curran Associates, Inc., 2023. 1, 2

[2] Sai Aparna Aketi, Sangamesh Kodge, and Kaushik Roy. Neighborhood gradient mean: An efficient decentralized learning method for non-IID data. *Transactions on Machine Learning Research*, 2023. 1, 2, 3, 4, 7

[3] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 4, 6

[4] Dan Alistarh, Torsten Hoefler, Mikael Johansson, Nikola Konstantinov, Sarit Khirirat, and Cedric Renggli. The convergence of sparsified gradient methods. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 4, 6

[5] Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *International Conference on Machine Learning*, pages 639–668. PMLR, 2022. 3

[6] Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Mike Rabbat. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, pages 344–353. PMLR, 2019. 1

[7] Aditya Balu, Zhanhong Jiang, Sin Yong Tan, Chinmay Hedge, Young M Lee, and Soumik Sarkar. Decentralized deep learning using momentum-accelerated consensus. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3675–3679, 2021. 1

[8] Debora Caldarola, Barbara Caputo, and Marco Ciccone. Improving generalization in federated learning by seeking flat minima. In *European Conference on Computer Vision*, pages 654–672. Springer, 2022. 3

[9] Rong Dai, Xun Yang, Yan Sun, Li Shen, Xinmei Tian, Meng Wang, and Yongdong Zhang. Fedgamma: Federated learning with global sharpness-aware minimization. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2023. 3

[10] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, et al. Large scale distributed deep networks. *Advances in neural information processing systems*, 25, 2012. 1

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[12] Jiawei Du, Zhou Daquan, Jiashi Feng, Vincent Tan, and Joey Tianyi Zhou. Sharpness-aware training for free. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 3, 8

[13] Yasaman Esfandiari, Sin Yong Tan, Zhanhong Jiang, Aditya Balu, Ethan Herron, Chinmay Hegde, and Soumik Sarkar. Cross-gradient aggregation for decentralized learning from non-iid data. In *International Conference on Machine Learning*, pages 3036–3046. PMLR, 2021. 1, 2, 3

[14] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. 2, 3, 8

[15] Noah Golmant, Amir Gholami, Michael Mahoney, and Joseph Gonzalez. pytorch-hessian-eigenthings: efficient pytorch hessian eigendecomposition, Oct. 2018. 8

[16] Robert Hönig, Yiren Zhao, and Robert Mullins. DAdaQuant: Doubly-adaptive quantization for communication-efficient federated learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8852–8866. PMLR, 17–23 Jul 2022. 1

[17] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip B. Gibbons. The non-iid data quagmire of decentralized machine learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020. 1

[18] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip B. Gibbons. The non-iid data quagmire of decentralized machine learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020. 5

[19] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019. 6

[20] Hamel Husain. Imagenette - a subset of 10 easily classified classes from the imagenet dataset. *https://github.com/fastai/imagenette*, 2018. 6

[21] Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho, and Krzysztof Geras. The break-even point on optimization trajectories of deep neural networks. *arXiv preprint arXiv:2002.09572*, 2020. 8

[22] Weisen Jiang, Hansi Yang, Yu Zhang, and James Kwok. An adaptive policy to employ sharpness-aware minimization. *arXiv preprint arXiv:2304.14647*, 2023. 3

[23] Yiding Jiang*, Behnam Neyshabur*, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020. 3

[24] Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian Stich, and Martin Jaggi. Error feedback fixes signsgd and other gradient compression schemes. In *International Conference on Machine Learning*, pages 3252–3261. PMLR, 2019. 6, 7

[25] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017. 3

[26] Anastasiia Koloskova, Tao Lin, and Sebastian U Stich. An improved analysis of gradient tracking for decentralized machine learning. *Advances in Neural Information Processing Systems*, 34:11422–11435, 2021. 1, 2, 3

[27] Anastasia Koloskova, Tao Lin, Sebastian U Stich, and Martin Jaggi. Decentralized deep learning with arbitrary communication compression. *arXiv preprint arXiv:1907.09356*, 2019. 1, 2, 3, 4

[28] Anastasia Koloskova, Sebastian Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3478–3487. PMLR, 09–15 Jun 2019. 4

[29] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. 2016. 1

[30] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar (canadian institute for advanced research). *http://www.cs.toronto.edu/ kriz/cifar.html*, 2014. 6

[31] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 5905–5914. PMLR, 2021. 3

[32] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 2

[33] 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. *Advances in Neural Information Processing Systems*, 30, 2017. 1, 2, 3, 5, 8

[34] Tao Lin, Sai Praneeth Karimireddy, Sebastian Stich, and Martin Jaggi. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6654–6665. PMLR, 18–24 Jul 2021. 1, 2, 3, 4, 5, 6

[35] Hanxiao Liu, Andy Brock, Karen Simonyan, and Quoc Le. Evolving normalization-activation layers. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13539–13550. Curran Associates, Inc., 2020. 5

[36] Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12360–12370, 2022. 3, 8

[37] Peng Mi, Li Shen, Tianhe Ren, Yiyi Zhou, Xiaoshuai Sun, Rongrong Ji, and Dacheng Tao. Make sharpness-aware minimization stronger: A sparsified perturbation approach. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 3

[38] Xinchi Qiu, Titouan Parcollet, Javier Fernandez-Marques, Pedro Porto Buarque de Gusmao, Yan Gao, Daniel J. Beutel, Taner Topal, Akhil Mathur, and Nicholas D. Lane. A first look into the carbon footprint of federated learning, 2023. 1

[39] Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. Generalized federated learning via sharpness aware minimization. In *International conference on machine learning*, pages 18250–18280. PMLR, 2022. 3, 5

[40] Yifan Shi, Li Shen, Kang Wei, Yan Sun, Bo Yuan, Xueqian Wang, and Dacheng Tao. Improving the model consistency of decentralized federated learning. In *International Conference on Machine Learning*, pages 31269–31291. PMLR, 2023. 3, 5

[41] Dongkuk Si and Chulhee Yun. Practical sharpness-aware minimization cannot converge all the way to optima. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 5

[42] Sebastian U. Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. NIPS'18, page 4452–4463, Red Hook, NY, USA, 2018. Curran Associates Inc. 4, 6

[43] Yan Sun, Li Shen, Shixiang Chen, Liang Ding, and Dacheng Tao. Dynamic regularized sharpness aware minimization in federated learning: Approaching global consistency and smooth landscape. In *International Conference on Machine Learning*, pages 32991–33013. PMLR, 2023. 3

[44] Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 7663–7673, Red Hook, NY, USA, 2018. Curran Associates Inc. 2

[45] Hanlin Tang, Xiangru Lian, Shuang Qiu, Lei Yuan, Ce Zhang, Tong Zhang, and Ji Liu. Deepsqueeze: Decentralization meets error-compensated compression. *arXiv preprint arXiv:1907.07346*, 2019. 1, 2

[46] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. d^2: Decentralized training over decentralized data. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4848–4856. PMLR, 10–15 Jul 2018. 1, 2

[47] Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. Practical low-rank communication compression in decentralized deep learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14171–14181. Curran Associates, Inc., 2020. 1

[48] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004. 1, 3

[49] Haoyu Zhao, Boyue Li, Zhize Li, Peter Richtárik, and Yuejie Chi. BEER: Fast \$o(1/t)\$ rate for decentralized nonconvex optimization with communication compression. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 2, 3

[50] Yang Zhao, Hao Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. In *International Conference on Machine Learning*, pages 26982–26992. PMLR, 2022. 3, 5

[51] Tongtian Zhu, Fengxiang He, Kaixuan Chen, Mingli Song, and Dacheng Tao. Decentralized SGD and average-direction SAM are asymptotically equivalent. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 43005–43036. PMLR, 23–29 Jul 2023. 3