CONTROLLED DISAGREEMENT IMPROVES GENERALIZATION IN DECENTRALIZED TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Decentralized training is often regarded as inferior to centralized training because the consensus errors between workers are thought to undermine convergence and generalization, even with homogeneous data distributions. This work challenges this view by introducing decentralized SGD with Adaptive Consensus (DSGD-AC), which intentionally preserves non-vanishing consensus errors through a time-dependent scaling mechanism. We prove that these errors are not random noise but systematically align with the dominant Hessian subspace, acting as structured perturbations that guide optimization toward flatter minima. Across image classification and machine translation benchmarks, DSGD-AC consistently surpasses both standard DSGD and centralized SGD in test accuracy and solution flatness. Together, these results establish consensus errors as a useful implicit regularizer and open a new perspective on the design of decentralized learning algorithms.

1 INTRODUCTION

In large-scale deep learning, decentralized optimization, where workers exchange model parameters only with neighbors, reduces the overhead of global synchronization and avoids costly all-reduce communication (Abadi et al., 2016; Li et al., 2020). This neighbor-only exchange can substantially reduce communication overhead, latency, and single points of failure, making decentralized approaches attractive for geographically distributed systems (Dhasade et al., 2023; Gholami & Seferoglu, 2024) and even GPU clusters (Lian et al., 2017; Assran et al., 2019; Wang et al., 2025).

Despite its practical appeal in runtime efficiency, decentralized training methods such as Decentralized Stochastic Gradient Descent (DSGD) are conventionally viewed as suboptimal compared to centralized/synchronous SGD in terms of convergence and, importantly, generalization performance even with i.i.d. data distributions among workers. This gap is largely attributed to consensus errors — persistent discrepancies in the model parameters maintained by different workers (Alghunaim & Yuan, 2022; Zhu et al., 2022). Prior work has focused heavily on minimizing these consensus errors to close the gap. Massive efforts have been made to reduce consensus errors, which involve communication topologies (Ying et al., 2021; Takezawa et al., 2023) and algorithm designs (Pu & Nedić, 2021; Wang et al., 2019; Lin et al., 2021) to ensure decentralized training closely approximates centralized training.

However, the conventional perspective neglects the potential constructive role of consensus errors. Rather than detrimental noise, these discrepancies may serve as structured perturbations that facilitate exploration of flatter minima in the loss landscape — solutions known to correlate with superior generalization (Jiang et al., 2019). This insight draws inspiration from sharpness-aware minimization strategies (Foret et al., 2020; Bisla et al., 2022; Li et al., 2024b), which explicitly introduce curvature-aware perturbations to enhance model robustness and performance.

In this study, we challenge the conventional view by introducing Decentralized SGD with Adaptive Consensus (DSGD-AC), an algorithm that strategically preserves non-vanishing consensus errors through an adaptive, time-dependent scaling mechanism. We provide a theoretical analysis demonstrating that consensus errors align with the dominant subspace of the Hessian matrix, thereby inducing beneficial curvature-related perturbations from the global average. Notably, DSGD-AC incurs negligible additional computational overhead relative to standard SGD or DSGD and enjoys the superior runtime efficiency over SGD at the same time.

Comprehensive experiments reveal that DSGD-AC consistently surpasses both DSGD and centralized SGD in terms of test accuracy and the flatness of the attained minima. To the best of our knowledge, this work constitutes the first demonstration of decentralized training outperforming centralized approaches under optimal conditions by a clear margin.

The main contributions of this work are: (1) the proposal of DSGD-AC, an adaptive consensus algorithm that maintains theoretically-justified non-vanishing consensus errors at minimal computational expense, (2) a theoretical characterization of consensus error and its alignment with the dominant Hessian subspace, and (3) empirical validation of the superior generalization performance of DSGD-AC on typical deep learning tasks.

1.1 Related works

Canonical view about consensus errors The prevailing perspective on decentralized training is that it should approximate synchronous/centralized training as closely as possible. To mitigate discrepancies among local models caused by weakly connected networks, prior work has focused on tracking global information (Wang et al., 2019; Pu & Nedić, 2021; Yuan et al., 2021; Takezawa et al., 2022), enhancing communication topologies to improve convergence rates (Ying et al., 2021; Zhu et al., 2022; Takezawa et al., 2023), and more. In addition, several theoretical studies (Zhu et al., 2022; Alghunaim & Yuan, 2022) establish a theoretical connection between the connectivity of decentralized communication topologies and both convergence and generalization, demonstrating that weaker connectivity results in poorer outcomes on both fronts. In contrast, we demonstrate the potential advantages of the consensus error by identifying its correlation with the dominant Hessian subspace, and we propose DSGD-AC in which consensus errors can, in practice, outperform SGD in deep learning tasks.

Explorations beyond the canonical view As the canonical perspective dominates, the effort that has been made towards suggesting potential benefits of consensus errors is limited. Kong et al. (2021) conducts empirical studies aimed at identifying thresholds of consensus errors. Although they highlight advantages of consensus errors in certain phases, the regime where consensus errors exceed those of DSGD with a ring topology remains unexplored, and the consensus control scheme proposed in the work does not yield clear performance improvements. Zhu et al. (2023) offers a novel interpretation, framing consensus errors in DSGD as random perturbations around the global average, which are asymptotically equivalent to average-direction SAM (Bisla et al., 2022). Our work further identifies the intrinsic curvature-related property of the consensus errors, and, by proposing DSGD-AC, empirically demonstrates the superior potential of decentralized training over centralized training without being limited to the large-batch setting.

Explicit curvature-related perturbations improve generalization but are costly With the idea of taking the global average as the deployed model (Zhu et al., 2023), decentralized learning can be interpreted as the learning on the (virtual) global average with the workers as the perturbed global average. Sharpness-aware minimization (SAM) was first proposed by Foret et al. (2020) to improve the generalization of deep neural networks, and many variants (Kwon et al., 2021; Bisla et al., 2022; Liu et al., 2022; Li et al., 2024a; Luo et al., 2024) were developed to further improve SAM. However, to achieve the best performance, the algorithms typically require one or more additional rounds of gradient evaluations, which significantly increase the computational costs. Our work utilizes the potential of the consensus errors as free perturbations to enhance the generalization without introducing extra computation.

2 Problem settings and notations

Practical remarks on data distributions and the distributed data sampler Our work focuses on decentralized training in GPU clusters where the whole dataset can be easily accessed by all workers. This scenario is also widely studied in many other literature (Assran et al., 2019; Ying et al., 2021; Kong et al., 2021; Wang et al., 2025), and is important for improving the efficiency of large-scale distributed training. The common practice for the distributed data sampler (also used in our experiments) is to reshuffle the full dataset at the start of each epoch and partition it evenly across workers. The strategy ensures, in expectation, i.i.d. data distributions among workers.

Decentralized Optimization We denote the set of integers $\{1, 2, \cdots, k\}$ by $[k]$, the number of workers by $n \in \mathbb{N}^+$, and the dimension of model parameters by $d \in \mathbb{N}^+$. In the standard decentralized optimization setup with n workers, each worker $i \in [n]$ holds a local objective determined by its local dataset $\mathcal{D}_i$:

$$f_i(x) = \mathbb{E}_{s\sim\mathcal{D}_i}[f_i(x; s)], \tag{1}$$

and the optimization problem is the local losses evaluated on the local models with a hard consensus constraint:

$$\underset{\{x_1,x_2,\cdots,x_n\}}{\text{minimize}} \; F(x_1, \cdots, x_n) = \frac{1}{n}\sum_{i=1}^{n} f_i(x_i), \quad \text{subject to } x_i = x_j, \forall i, j \in [n] \tag{2}$$

In the i.i.d. data distribution setting, we have $f_i = f_j = F$ for all $i, j \in [n]$.

Decentralized SGD (DSGD) The update of DSGD (Lian et al., 2017) on worker i is:

$$x_i^{(t)} = x_i^{(t-1)} - \alpha^{(t)}\nabla f(x_i^{(t-1)}; s_i^{(t)}) + \sum_{j\in\mathcal{N}(i)} W_{ij}(x_j^{(t-1)} - x_i^{(t)}) \tag{3}$$

where $\mathcal{N}(i)$ is the set of neighbors of worker i (including itself), W is a symmetric, non-negative, and doubly stochastic matrix defining the weights of the edges ($W_{ij} = 0$ if worker i is not a neighbor of worker j), and $s_i^{(t)}$ denotes the stochastic mini-batch sampled by worker i at iteration t.

Following the common notations in decentralized optimization, we denote the global average by $\bar{x}^{(t)} := \sum_{i=1}^n x_i^{(t)}$, the consensus error of worker i by $e_i^{(t)} := x_i^{(t)} - \bar{x}^{(t)}$, the matrix form of all local models by $X^{(t)} := [x_1^{(t)}, \cdots, x_n^{(t)}] \in \mathbb{R}^{d\times n}$, the matrix form of all local stochastic gradients by $G^{(t)} := [\nabla f_1(x_1^{(t-1)}; s_1^{(t)}), \cdots, \nabla f_n(x_n^{(t-1)}; s_n^{(t)})]$, the matrix form of all consensus errors as $\Delta^{(t)} = [e_1^{(t)}, \cdots, e_n^{(t)}]$, and the matrix $\bar{X}$ by $\bar{X}^{(t)} = [\bar{x}^{(t)}, \cdots, \bar{x}^{(t)}]$.

Note that there is another variant of DSGD that performs the local update before communication. We focus on the variant in Eq. (3) as it is shown to be more efficient (Lian et al., 2017; Wang et al., 2025), and two variants are proven to have the same generalization bound (Bellet et al., 2023).

3 DSGD-AC: Decentralized SGD with adaptive consensus

In this section, we use the experiment of training a wide ResNet (WRN28-10) (Zagoruyko & Komodakis, 2016) on CIFAR-10 (Krizhevsky et al., 2009) as a showcase to demonstrate the limitation of DSGD and the improvement brought by our proposed algorithm. In the experiment, we employ the standard cosine annealing learning rate schedule (Loshchilov & Hutter, 2016) with a linear warm-up during the first 10 epochs (Figure 1, left). This learning rate schedule is commonly used in practice and can strike a balance between the training stability and generalization Gotmare et al. (2018); Kalra & Barkeshli (2024). For decentralized training, we use 8 workers and the one-peer ring as the decentralized communication topology.

3.1 Finding: vanishing consensus error in DSGD

We start by empirically investigating the dynamics of consensus errors when trained with DSGD. We track the average norm of the consensus errors during the training. We observe that, for DSGD, the consensus errors gradually vanish as the learning rate decreases (Figure 1, right).

From the theoretical perspective, by interpreting the mixing step as a gradient step on a quadratic consensus penalty, one obtains the per-step surrogate

$$\begin{aligned} &J^{(t)}(x_1, \cdots, x_n) \\ &= \sum_{i=1}^{n} f_i(x_i^{(t)}) + \frac{1}{2\alpha^{(t)}} \sum_{i,j\in[n]} W_{ij}\|x_i^{(t)} - x_j^{(t)}\|^2 \\ &= \underbrace{\sum_{i=1}^{n} f_i(\bar{x}^{(t)})}_{\text{objective on deployed model}} + \underbrace{\sum_{i=1}^{n}[f_i(x_i^{(t)}) - f_i(\bar{x}^{(t)})]}_{\text{sharpness}} + \underbrace{\frac{1}{2\alpha^{(t)}} \sum_{i,j\in[n]} W_{ij}\|x_i^{(t)} - x_j^{(t)}\|^2}_{\text{consensus regularizer}} \end{aligned} \tag{4}$$

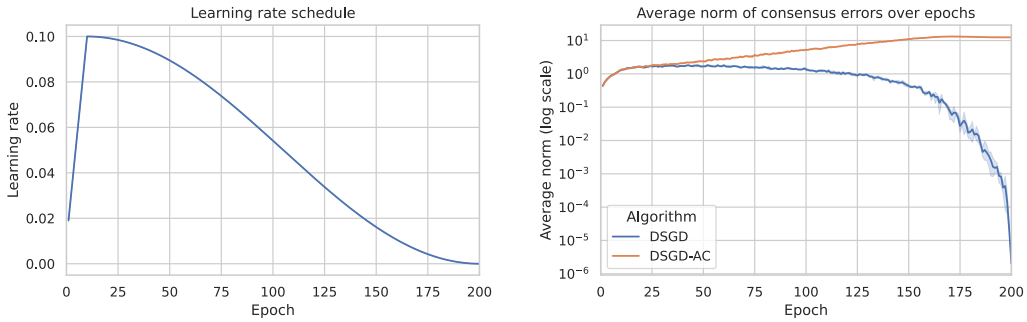

Figure 1: Decentralized training of WRN28-10 on CIFAR-10 (3 random runs for each algorithm) with 8 workers, and the communication topology is the one-peer ring topology. **Left**: Learning rate schedule (same for both algorithms). **Right**: Average norm of consensus errors evaluated at the end of every epoch ($\frac{1}{N}\sum_{i=1}^{N}\|x_i^{(eT)} - \bar{x}^{(eT)}\|$). p is set to 3 for DSGD-AC.

With symmetric mixing weights and no momentum or adaptivity, each DSGD step is exactly a (stochastic) gradient on J. Thus, when $\alpha^{(t)}$ goes to 0, the consensus regularizer dominates the objective function, which minimizes the consensus errors. If considering this surrogate function, the empirical observation is not surprising because it reflects the hard constraint in the optimization problem in Eq. (2). However, the vanishing consensus errors void the sharpness term in Eq. (4) because the sharpness term because $f_i(x_i^{(t)}) \approx f_i(\bar{x}^{(t)})$ as $x_i^{(t)} - \bar{x} \to 0$. The only term left that is relevant to the deployed model $\bar{x}^{(t)}$ is the first term, which is the same objective as in synchronous SGD. Therefore, to maintain the potential benefits of free sharpness-aware regularization (Zhu et al., 2023) by the consensus errors, we need to maintain a non-vanishing radius throughout the training.

3.2 Algorithm: Decentralized SGD with adaptive consensus

The proposed algorithm is shown in Algorithm 1. The difference from DSGD is highlighted, and, compared with DSGD, the proposed variant includes an adaptive factor to maintain non-diminishing consensus errors intentionally. At the end of training, the algorithm takes the global average of all local models as the deployed model.

Algorithm 1: Decentralized SGD with adaptive consensus (DSGD-AC) on worker i

Data: Dataset (D), the number of workers (N), the number of epoch (E), the number of batches per epoch (T), intialization ($x^{(0)}$), and a hyperparameter ($p \in \mathbb{R}^+$).

Result: Deployed model $\bar{x} = \frac{1}{n}\sum_{j=1}^{n} x_j^{(TE)}$

$x_1^{(0)} = x_2^{(0)} = \cdots = x_n^{(0)} = x^{(0)}$

for $t = 1$ *to* TE **do**

$g_i^{(t)} = \nabla f(x_i^{(t-1)}; s_i^{(t)})$

$\gamma^{(t)} = \left[\alpha^{(t)}/\alpha_{\max}\right]^p$

$x_i^{(t)} = x_i^{(t-1)} - \alpha^{(t)} g_i^{(t)} + \gamma^{(t)} \sum_{j\in\mathcal{N}(i)} W_{ij}(x_j^{(t-1)} - x_i^{(t-1)})$

end

Note that $\alpha^{(t)}$ is determined by the learning rate scheduler like cosine annealing (Loshchilov & Hutter, 2016), and $\alpha_{\max}$ is the maximal learning rate throughout the training, which ensures $\gamma^{(t)}$ is in the range $[0, 1]$.

We evaluate the performance of DSGD-AC on classical deep learning tasks in Section 4. In the numerical experiments, the results demonstrate the superior generalization performance of DSGD-

AC over DSGD and centralized SGD. We will analyze the reasons behind this by showing that DSGD-AC maintains non-diminishing and useful consensus errors in the following sections.

3.3 CONTROLLED CONSENSUS ERRORS IN DSGD-AC

The motivation of DSGD-AC is to maintain non-diminishing consensus errors. Therefore, we multiply the weight of the consensus regularizer in Eq. (4) by an adaptive γ, which directly leads to the DSGD-AC algorithm. The per-step surrogate function of DSGD-AC is mostly the same as that of DSGD. Only the weight of the consensus regularizer becomes $\gamma^{(t)}/(2N\alpha^{(t)})$.

In this section, we investigate the impact of p on the magnitude of consensus errors. First, we can rewrite the update of DSGD-AC in matrix form,

$$X^{(t)} = X^{(t-1)} - \alpha^{(t)}G^{(t)} - \gamma^{(t)}X^{(t-1)}(I-W) = X^{(t-1)}(I-\gamma^{(t)}L) - \alpha^{(t)}G^{(t)} \tag{5}$$

where we denote the Laplacian matrix L by $L = I - W$.

By subtracting $\bar{X}^{(t)}$ on both sides of Eq. (5) and using the fact that $\Delta^{(t)} = (I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)X^{(t)}$, the dynamics of consensus errors $\Delta^{(t)}$ can be derived as

$$\begin{aligned}
\Delta^{(t)} &= X^{(t-1)}(I-\gamma^{(t)}L) - \alpha^{(t)}G^{(t)} - \bar{X}^{(t)} \\
&= X^{(t-1)}(I-\gamma^{(t)}L) - \alpha^{(t)}G^{(t)} - \bar{X}^{(t-1)} + \alpha^{(t)}G^{(t)}\cdot\frac{1}{n}\mathbf{1}\mathbf{1}^\top \\
&= \Delta^{(t-1)}(I-\gamma^{(t)}L) - \alpha^{(t)}G^{(t)}(I-\frac{1}{n}\mathbf{1}\mathbf{1}^\top)
\end{aligned} \tag{6}$$

Next, we denote $P = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$, perform an eigen-decomposition of $L = U\Lambda U^\top$, and multiply Eq. (6) by U from the right to obtain

$$\begin{aligned}
\Delta^{(t)}U &= \Delta^{(t-1)}(I-\gamma^{(t)}U\Lambda U^\top)U - \alpha^{(t)}G^{(t)}PU \\
&= \Delta^{(t-1)}U(I-\gamma^{(t)}\Lambda) - \alpha^{(t)}G^{(t)}PU
\end{aligned} \tag{7}$$

By introducing $Z^{(t)} = \Delta^{(t)}U$ and $\tilde{G}^{(t)} = G^{(t)}PU$, we can re-write the update as

$$Z^{(t)} = Z^{(t-1)}(I-\gamma^{(t)}\Lambda) - \alpha^{(t)}\tilde{G}^{(t-1)} \tag{8}$$

Here, $Z^{(t)}$ represents the consensus error projected onto the eigen-basis of the Laplacian. Each component of $Z^{(t)}$ quantifies the magnitude of consensus errors along a distinct network mode, capturing how different patterns of agent disagreement evolve and persist according to the network topology and communication structure. By analyzing the dynamics of $Z^{(t)}$ we derive the following proposition with the proof deferred to Section A.1 in the appendix.

Proposition 1 (p for non-vanishing consensus errors) *In a quasi-stationary regime where the expectations and variances of the columns of $\tilde{G}^{(t)}$ remain constant, DSGD-AC with $p \geq 2$ ensures that the expected Frobenius norm of $\Delta^{(t)}$ does not vanish even as $\alpha^{(t)} \to 0$.*

The proposition establishes that DSGD-AC maintains a nontrivial level of consensus errors throughout iterations. In fact, the proof of the proposition shows that the effective disagreement radius $r_t^2 = \mathbb{E}[\|\Delta^{(t)}\|_F^2]$ is on the order of $(\alpha^{(t)})^2/\gamma^{(t)}$. Since it has been empirically observed (see, *e.g.*, Bisla et al. (2022); Li et al. (2024a)) that it is advantageous to increase the radius slightly towards the end of the training, we chose $\gamma^{(t)} = g_0(\alpha^{(t)})^p$ with $p = 3$. Under cosine learning rate schedules, this choice induces a mild uptick in the radius toward the final stages of training, as illustrated in Figure 1 (right). A detailed sensitivity analysis in Appendix A.4.2 further supports the theory, demonstrating radius shrinkage for $p < 2$ and growth for $p > 2$ as $\alpha^{(t)} \to 0$ (see Figure 10).

3.4 CONSENSUS ERRORS ALIGN WITH DOMINANT SUBSPACE OF HESSIAN

Even though DSGD-AC maintains non-vanishing consensus errors, its role in leading to flatter minima and better generalization remains underexplored. In (Zhu et al., 2023, Theorem 1), consensus

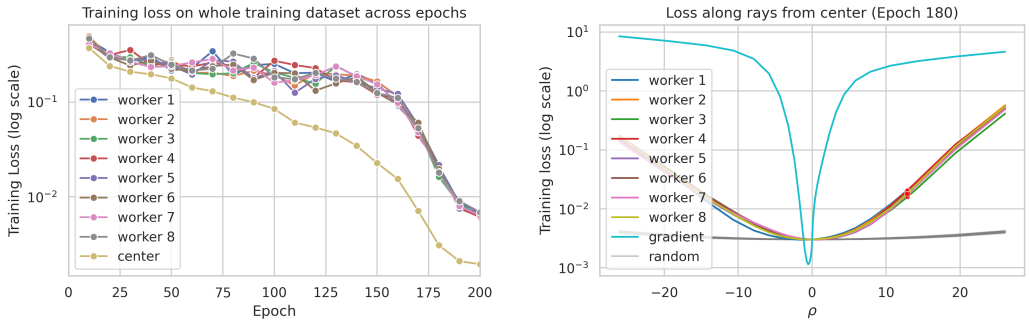

Figure 2: **Left**: Losses on the whole training dataset at local workers and global average. The losses are evaluated every 10 epochs. **Right**: Training loss at epoch 180 along: (1) `worker i`: lines connecting global average and worker i, (2) `gradient`: the line that aligns with the full-batch gradient at the global average and crosses the global average, and (3) `random`: 500 lines that cross the global average and follow random directions generated as in (Bisla et al., 2022). The x-axis means the directional magnitude of the perturbation along these directions. The red dots represent the losses at the local models. The losses are computed on $\sim 1/4$ of the training dataset due to computation complexity.

errors are interpreted as random perturbations within the subspace defined by the weight diversity matrix, and the resulting (asymptotically equivalent) average-direction SAM effect is shown to improve generalization. While this connection is insightful, the intrinsic structure of consensus errors (or the weight diversity matrix) has not been examined in detail.

To study this structure, we compare the training losses at local models with those at their global average. As shown in Figure 2 (left), the global average consistently achieves lower training loss than any individual worker. To further distinguish consensus errors from random perturbations, we evaluate the training losses along the directions of consensus errors and compare them against losses along sufficiently many random directions. Figure 2 (right) shows that the random directions are almost flat, which is expected given the large parameter space ($\sim$36M in WRN28-10) and the low-rank structure of the Hessian (Gur-Ari et al., 2018; Song et al., 2024). It is also consistent with empirical observations in (Keskar et al., 2016).

In contrast, directions induced by consensus errors yield significant increases in training loss, highlighting that these errors are not random but aligned with directions of higher curvature. This phenomenon suggests a correlation between consensus errors and the dominant subspace of the Hessian (or directions with larger curvature). Motivated by this observation, we formalize it in the following proposition, with the proof deferred to Appendix A.2.

Proposition 2 (Consensus error aligns with dominant Hessian subspace) *Let x^* be a locally strongly convex minimizer of F, with Hessian $H = U\Lambda U^\top$ and eigenvalues $0 < \lambda_1 \leq \cdots \leq \lambda_d$. Consider the linearized DSGD-AC dynamics around x^* under i.i.d. local data distributions ($f_i = f$), a symmetric doubly stochastic communication matrix W, and non-increasing stepsizes $\alpha^{(t)} > 0$. For each Hessian eigenvector u_k ($k \in [d]$), the norm of the projected consensus error $\Delta_k^{(t)} := u_k^\top \Delta^{(t)}$ evolves as a linear system with stability condition*

$$\alpha^{(t)} < \frac{2 + (\lambda_{\min}^W - 1)\gamma^{(t)}}{\lambda_k}.$$

Therefore, modes with smaller λ_k stabilize earlier, while modes with larger λ_k retain higher steady-state variance. As a result, the consensus error energy concentrates on the subspace spanned by the top eigenvectors of H, i.e., the dominant Hessian subspace.

Given the alignment between the consensus errors and the dominant subspace, DSGD-AC can be interpreted as optimization over $\bar{x}$ with curvature-correlated noises, which has been both empirically and theoretically studied by many works (Foret et al., 2020; Zhang et al., 2023; Luo et al.,

2024; Benedetti & Ventura, 2024). By maintaining non-vanishing consensus errors along with its regularization effect on the curvature of the loss landscape, DSGD-AC is expected to achieve better generalization performance than DSGD and SGD.

While the alignment exists and can be shown theoretically, the alignment is noisy and spans on less-sharp directions when compared with the gradient direction. As shown in Figure 2 (right), the gradient computed on the corresponding dataset leads to a sharper increase than the consensus errors. An interesting direction for future work could be an improved algorithm based on DSGD-AC that can utilize the gradient information to promote the concentration of consensus errors on the dominant Hessian subspace with small computational overhead.

4 Numerical Experiments

In this section, we present the results of the numerical experiments on image classification with wide ResNet (Zagoruyko & Komodakis, 2016) and on machine translation with transformers (Vaswani et al., 2017). In the experiments, we follow hyperparameters in the corresponding original papers, and we reproduce the same baseline performance for a fair comparison. For DSGD-AC, we use $p = 3$ in all experiments. We defer the other hyperparameter details to Appendix A.3 and the sensitivity analysis on p to Appendix A.4.2.

Each set of experiments consists of three random runs with fixed random seeds. We report $1\times$ standard deviation in all tables, and the shaded areas in plots correspond to the 95% confidence interval.

4.1 Image classification with wide ResNet

We train two variants of Wide ResNets (WRN28-10 and WRN16-8) (Zagoruyko & Komodakis, 2016) on two datasets, CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009), and we present the curves of the classification accuracies on the test set and training/test losses of WRN28-10 on CIFAR-10 in Figure 3. The test performance and the flatness of the solutions are reported in Table 4.1. The curves and statistics of the remaining experiments are deferred to Appendix A.4.1.

Since finding the best sharpness metric that always reflects the potential generalization is still an open question, we use the top-1 eigenvalue as a surrogate, which is widely used in other literature and proven to have a strong correlation (Bisla et al., 2022; Luo et al., 2024).

Model	Dataset	Algorithm	Test Acc. (%) ↑	Test Loss ↓	Top-1 Eigenvalue ↓
WRN28-10	CIFAR-10	DSGD	96.07 ± 0.13	0.176 ± 0.005	22.43 ± 3.99
		SGD	95.96 ± 0.14	0.182 ± 0.004	16.84 ± 0.32
		DSGD-AC	**96.77 ± 0.11**	**0.128 ± 0.003**	**8.96 ± 0.35**
	CIFAR-100	DSGD	79.86 ± 0.22	0.899 ± 0.008	49.57 ± 4.80
		SGD	80.15 ± 0.42	0.878 ± 0.020	37.37 ± 2.88
		DSGD-AC	**82.38 ± 0.09**	**0.755 ± 0.008**	**19.80 ± 0.66**
WRN16-8	CIFAR-10	DSGD	95.94 ± 0.11	0.152 ± 0.001	18.19 ± 0.64
		SGD	95.81 ± 0.13	0.153 ± 0.003	17.49 ± 1.61
		DSGD-AC	**96.17 ± 0.04**	**0.129 ± 0.003**	**11.82 ± 0.48**
	CIFAR-100	DSGD	79.25 ± 0.26	0.854 ± 0.016	36.19 ± 3.80
		SGD	79.42 ± 0.18	0.849 ± 0.015	33.77 ± 0.78
		DSGD-AC	**80.67 ± 0.11**	**0.771 ± 0.005**	**19.81 ± 0.16**

Table 1: Performance comparison of DSGD, SGD, and DSGD-AC on image classification with wide ResNet. The top-1 eigenvalue is computed on the whole training set and approximated using the power iteration.

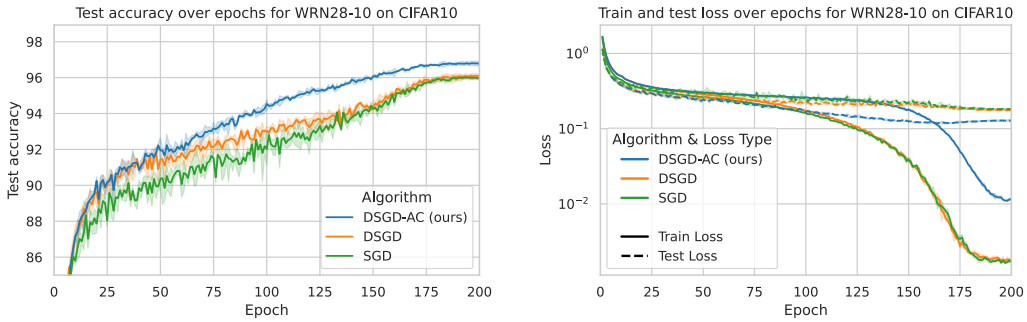

Figure 3: WRN28-10 on CIFAR-10. **Left**: Test accuracy on test set. For decentralized training, the accuracy is evaluated on the global average model. **Right**: Training losses (evaluated on the workers for decentralized training, and evaluated on perturbed points for SAM) and test losses (evaluated on the global average model for decentralized training). The curves for each algorithm are based on 3 runs with the same set of random seeds.

In the experiment results, DSGD-AC outperforms DSGD and SGD in test accuracy, test losses, and solution flatness by a clear margin. Moreover, it can also be seen that DSGD can not outperform SGD with its best performance.

4.2 Machine translation with transformers

We also validate the idea of controlling consensus errors on transformer models by simply replacing the local update with the Adam optimizer (Adam et al., 2014). DSGD-AC is then adapted to DAdam-AC. We train Transformer (the big variant, $\sim$213M parameters) (Vaswani et al., 2017) on WMT14 (English-to-German) (Bojar et al., 2014) and present the curves of training losses and BLEU scores on the test set. The BLEU scores (Papineni et al., 2002) (which is used to evaluate the translation quality in the original paper) and the losses on the test set and the training set are reported in Table 2.

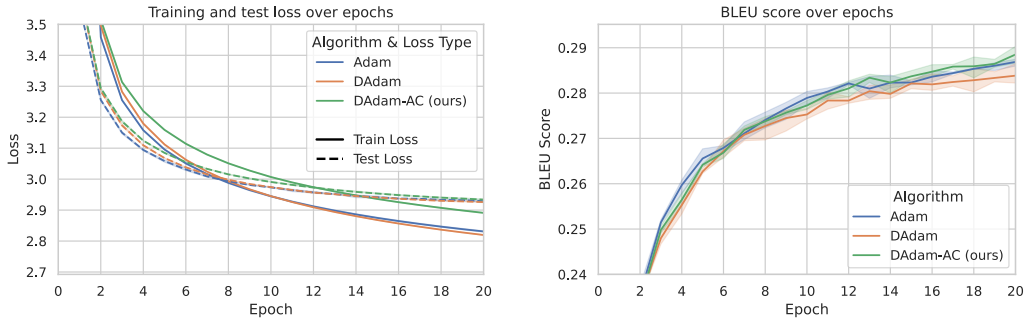

Figure 4: Transformer (big) on WMT14 English-to-German. **Left**: Losses on training set. **Right**: BLEU scores on the test set.

Algorithm	BLEU score ↑	Test loss ↓	Train loss ↓
Adam	28.68 ± 0.07	2.9290 ± 0.0026	2.8310 ± 0.0019
DAdam	28.38 ± 0.22	**2.9258 ± 0.0018**	**2.8195 ± 0.0008**
DAdam-AC	**28.85 ± 0.18**	2.9338 ± 0.0017	2.8909 ± 0.0012

Table 2: Performance comparison of DAdam, Adam, and DAdam-AC on neural machine translation with the transformer model.

The results demonstrate that DAdam-AC can outperform other baselines on the translation quality metric. The adaptive consensus brings substantial improvement compared with DAdam. We believe further improvement is possible if we take the adaptive consensus into account when designing the optimizer.

5 CONCLUSION

This work challenges the long-standing perception that decentralized training inevitably sacrifices generalization for communication efficiency. Through DSGD-AC, we demonstrate that maintaining controlled consensus errors improves robustness and solution flatness, offering both practical scalability and superior model performance. The method introduces negligible computational overhead and integrates seamlessly with existing decentralized frameworks. Our experiments on CIFAR benchmarks and WMT14 confirm the broad applicability of this approach. These results suggest a paradigm shift: consensus errors should no longer be minimized at all costs but strategically managed as a form of implicit regularization. Beyond immediate applications to deep learning clusters, we envision that the principle of adaptive consensus could inform the design of future large-scale, resource-efficient, and generalizable learning systems.

REFERENCES

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. TensorFlow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pp. 265–283, 2016.

Kingma DP Ba J Adam et al. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 1412(6), 2014.

Sulaiman A Alghunaim and Kun Yuan. A unified and refined convergence analysis for non-convex decentralized learning. *IEEE Transactions on Signal Processing*, 70:3264–3279, 2022.

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

Aurélien Bellet, Marc Tommasi, Kevin Scaman, Giovanni Neglia, et al. Improved stability and generalization guarantees of the decentralized SGD algorithm. In *Forty-first International Conference on Machine Learning*, 2023.

Marco Benedetti and Enrico Ventura. Training neural networks with structured noise improves classification and generalization. *Journal of Physics A: Mathematical and Theoretical*, 57(41): 415001, 2024.

Devansh Bisla, Jing Wang, and Anna Choromanska. Low-pass filtering SGD for recovering flat optima in the deep learning optimization landscape. In *International Conference on Artificial Intelligence and Statistics*, pp. 8299–8339. PMLR, 2022.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pp. 12–58, 2014.

Aaron Defazio, Xingyu Yang, Ahmed Khaled, Konstantin Mishchenko, Harsh Mehta, and Ashok Cutkosky. The road less scheduled. *Advances in Neural Information Processing Systems*, 37: 9974–10007, 2024.

Akash Dhasade, Anne-Marie Kermarrec, Rafael Pires, Rishi Sharma, and Milos Vujasinovic. Decentralized learning made easy with decentralizepy. In *Proceedings of the 3rd Workshop on Machine Learning and Systems*, pp. 34–41, 2023.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Peyman Gholami and Hulya Seferoglu. Digest: Fast and communication efficient decentralized learning with local updates. *IEEE Transactions on Machine Learning in Communications and Networking*, 2:1456–1474, 2024.

Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. *arXiv preprint arXiv:1810.13243*, 2018.

Guy Gur-Ari, Daniel A Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

Dayal Singh Kalra and Maissam Barkeshli. Why warmup the learning rate? underlying mechanisms and improvements. *Advances in Neural Information Processing Systems*, 37:111760–111801, 2024.

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. *arXiv preprint arXiv:1609.04836*, 2016.

Lingjing Kong, Tao Lin, Anastasia Koloskova, Martin Jaggi, and Sebastian Stich. Consensus control for decentralized deep learning. In *International Conference on Machine Learning*, pp. 5686–5696. PMLR, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. ASAM: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International conference on machine learning*, pp. 5905–5914. PMLR, 2021.

Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, et al. Pytorch distributed: Experiences on accelerating data parallel training. *arXiv preprint arXiv:2006.15704*, 2020.

Tao Li, Qinghua Tao, Weihao Yan, Zehao Lei, Yingwen Wu, Kun Fang, Mingzhen He, and Xiaolin Huang. Revisiting random weight perturbation for efficiently improving generalization. *arXiv preprint arXiv:2404.00357*, 2024a.

Tao Li, Pan Zhou, Zhengbao He, Xinwen Cheng, and Xiaolin Huang. Friendly sharpness-aware minimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5631–5640, 2024b.

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.

Tao Lin, Sai Praneeth Karimireddy, Sebastian U Stich, and Martin Jaggi. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. *arXiv preprint arXiv:2102.04761*, 2021.

Yong Liu, Siqi Mai, Minhao Cheng, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Random sharpness-aware minimization. *Advances in neural information processing systems*, 35:24543–24556, 2022.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Haocheng Luo, Tuan Truong, Tung Pham, Mehrtash Harandi, Dinh Phung, and Trung Le. Explicit eigenvalue regularization improves sharpness-aware minimization. *Advances in Neural Information Processing Systems*, 37:4424–4453, 2024.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Shi Pu and Angelia Nedić. Distributed stochastic gradient tracking methods. *Mathematical Programming*, 187(1):409–457, 2021.

Minhak Song, Kwangjun Ahn, and Chulhee Yun. Does SGD really happen in tiny subspaces? *arXiv preprint arXiv:2405.16002*, 2024.

Yuki Takezawa, Han Bao, Kenta Niwa, Ryoma Sato, and Makoto Yamada. Momentum tracking: Momentum acceleration for decentralized deep learning on heterogeneous data. *arXiv preprint arXiv:2209.15505*, 2022.

Yuki Takezawa, Ryoma Sato, Han Bao, Kenta Niwa, and Makoto Yamada. Beyond exponential graph: Communication-efficient topologies for decentralized learning via finite-time convergence. *Advances in Neural Information Processing Systems*, 36:76692–76717, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. SlowMo: Improving communication-efficient distributed SGD with slow momentum. *arXiv preprint arXiv:1910.00643*, 2019.

Zesen Wang, Zhang Jiaojiao, Wu Xuyang, and Mikael Johansson. From promise to practice: realizing high-performance decentralized training. In *The Thirteenth International Conference on Learning Representations*. ICLR, 2025.

Bicheng Ying, Kun Yuan, Yiming Chen, Hanbin Hu, Pan Pan, and Wotao Yin. Exponential graph is provably efficient for decentralized deep training. *Advances in Neural Information Processing Systems*, 34:13975–13987, 2021.

Kun Yuan, Yiming Chen, Xinmeng Huang, Yingya Zhang, Pan Pan, Yinghui Xu, and Wotao Yin. DecentLaM: Decentralized momentum SGD for large-batch deep training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3029–3039, 2021.

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

Hongyang R Zhang, Dongyue Li, and Haotian Ju. Noise stability optimization for finding flat minima: A Hessian-based regularization approach. *arXiv preprint arXiv:2306.08553*, 2023.

Tongtian Zhu, Fengxiang He, Lan Zhang, Zhengyang Niu, Mingli Song, and Dacheng Tao. Topology-aware generalization of decentralized SGD. In *International Conference on Machine Learning*, pp. 27479–27503. PMLR, 2022.

Tongtian Zhu, Fengxiang He, Kaixuan Chen, Mingli Song, and Dacheng Tao. Decentralized SGD and average-direction SAM are asymptotically equivalent. In *International Conference on Machine Learning*, pp. 43005–43036. PMLR, 2023.

A Appendix

A.1 Proof of Proposition 1

Recall that

$$\begin{aligned} P &= I - \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top \\ L &= I - W = U\Lambda U^\top \\ \tilde{G}^{(t)} &= G^{(t)}PU \\ Z^{(t)} &= Z^{(t-1)}(I - \gamma^{(t)}\Lambda) - \alpha^{(t)}\tilde{G}^{(t)} \end{aligned} \tag{9}$$

where $Z^{(t)}$ describes the consensus error projected onto the eigenbasis of the Laplacian.

Each column $z_k^{(t)}$ of $Z^{(t)}$ evolves as

$$z_k^{(t)} = (1 - \gamma^{(t)}\lambda_k)z_k^{(t-1)} - \alpha^{(t)}\tilde{g}_k^{(t)} \tag{10}$$

where λ_k is the k-th eigenvalue of L.

To quantify the dynamics of $\|z_k^{(t)}\|_2^2$, consider a quasi-stationary regime where

$$\mathbb{E}[\tilde{g}_i^{(t)}] = \mu_i, \quad \mathbb{E}[\|\tilde{g}_i^{(t)} - \mu_i\|_2^2] = \sigma_i^2 \tag{11}$$

Then, by taking the expectation on Eq. (10) and letting $m_i = \mathbb{E}[z_i]$, we have

$$m_i = (1 - \gamma^{(t)}\lambda_i)m_i - \alpha^{(t)}\mu_i \tag{12}$$

from which we find (for all modes $i \geq 2$)

$$m_i = -\frac{1}{\lambda_i}\frac{\alpha^{(t)}}{\gamma^{(t)}}\mu_i \tag{13}$$

Next, we define $\tilde{z}_i^{(t)} = z_i^{(t)} - m_i$ so that

$$\tilde{z}_i^{(t)} = (1 - \gamma^{(t)}\lambda_i)\tilde{z}_i^{(t-1)} - \alpha^{(t)}(\tilde{g}_i^{(t-1)} - \mu_i)$$

where we have subtracted m_i from both sides and used the expression for m_i just derived. Letting $V_k = \mathbb{E}[\|\tilde{z}_k^{(t)}\|_2^2]$ and assuming that the innovation $\eta^{(t-1)} = \tilde{g}_i^{(t-1)} - \mu_i$ is conditionally independent given all events generated up until iteration $t-1$, we find

$$V_i = (1 - \gamma^{(t)}\lambda_i)^2 V_i + (\alpha^{(t)})^2\sigma_i^2$$

which implies that

$$V_i = \frac{\left(\alpha^{(t)}\right)^2}{2\lambda_i\gamma^{(t)} - \lambda_i^2(\gamma^{(t)})^2}\sigma_i^2 \geq \frac{(\alpha^{(t)})^2}{2\lambda_i\gamma^{(t)}}\sigma_i^2.$$

Now, putting the two expression together yields

$$\mathbb{E}[\|z_i^{(t)}\|_2^2] = V_i + \|m_i\|_2^2 \geq \frac{1}{\lambda_i^2}\left(\frac{\alpha^{(t)}}{\gamma^{(t)}}\right)^2\|\mu_i\|_2^2 + \frac{1}{2\lambda_i}\frac{(\alpha^{(t)})^2}{\gamma^{(t)}}\sigma_i^2$$

Hence, with $\gamma^{(t)} = g_0(\alpha^{(t)})^p$, we have

$$\mathbb{E}[\|z_i^{(t)}\|_2^2] \geq \frac{1}{\lambda_i^2 g_0^2}(\alpha^{(t)})^{2-2p} + \frac{1}{2\lambda_i g_0}(\alpha^{(t)})^{2-p}$$

The result indicates that to maintain a non-diminishing $\|z_i\|_2^2$ in this quasi-stationary regime, we should choose $p \geq 2$. Finally, since $\|Z^{(t)}\|_F^2 = \mathrm{Tr}((\Delta^{(t)}U)^\top\Delta^{(t)}U) = \mathrm{Tr}((\Delta^{(t)})^\top\Delta^{(t)}) = \|\Delta^{(t)}\|_F^2$, a non-diminishing $\|Z^{(t)}\|_F^2$ implies that $\|\Delta^{(t)}\|_F^2$ is non-diminishing.

A.2 Proof of Proposition 2

In this proof, we focus on a local minima x^* of $F(x)$ that is locally strongly convex, and we denote $X^* = [x^*, \cdots, x^*] \in \mathbb{R}^{d\times n}$.

Rewrite the update of DSGD-AC by replacing the stochastic gradient with its expectation and a noise matrix,

$$X^{(t)} = X^{(t-1)}(I - \gamma^{(t)}L) - \alpha^{(t)}(\nabla F(X^{(t-1)}) + \Xi^{(t)}) \tag{14}$$

where

$$\begin{aligned} \nabla F(X^{(t-1)}) &:= [\nabla f_1(x_1), \cdots, \nabla f_n(x_n)] \\ \Xi^{(t)} &:= G^{(t)} - \nabla F(X^{(t-1)}) \end{aligned} \tag{15}$$

Then, in a similar way as in Section 3.3, we can rewrite the update of $\Delta^{(t)}$ as

$$\Delta^{(t)} = \Delta^{(t-1)}(I - \gamma^{(t)}L) - \alpha^{(t)}\nabla F(X^{(t-1)})P - \alpha^{(t)}\Xi^{(t)}P \tag{16}$$

By further assuming the local data distributions are i.i.d., we have $f_i = F$ and $H_i = H$ for all $i \in [n]$. By doing Taylor's expansion on x^* and ignoring the higher-order terms, we have

$$\nabla f_i(x_i^{(t-1)}) \approx \nabla f_i(x^*) + H_i(x_i^{(t-1)} - x^*) = H(x_i^{(t-1)} - x^*) \tag{17}$$

then we have

$$\nabla F(X^{(t-1)}) \approx H(X^{(t-1)} - X^*) \Rightarrow \nabla F(X^{(t-1)})P \approx H\Delta^{(t-1)} \tag{18}$$

Therefore, by plugging Eq. (18) back to Eq. (16), we have

$$\Delta^{(t)} \approx \Delta^{(t-1)}(I - \gamma^{(t)}L) - \alpha^{(t)}H\Delta^{(t-1)} - \alpha^{(t)}\Xi^{(t)}P \tag{19}$$

Take the eigen decomposition of H, we have $H = U\Lambda U^\top$. We denote the k-th eigenvector of H by u_k, the projection of consensus error on u_k by $\Delta_k^{(t)} := u_k^\top \Delta^{(t)}$, and the projection of gradient noise by $\xi_k^{(t)} = u_k^\top \Xi^{(t)} P$.

By multiplying $u_k^\top$ on both sides of Eq. (19) from the left, we have the dynamics of the projection of the consensus errors on u_k as

$$\begin{aligned} \Delta_k^{(t)} &\approx \Delta_k^{(t-1)}(I - \gamma^{(t)}L) - \alpha^{(t)}\lambda_k\Delta_k^{(t-1)} - \alpha^{(t)}\Xi_k^{(t)} \\ &= \Delta_k^{(t-1)}(I - \gamma^{(t)}L - \alpha^{(t)}\lambda_k I) - \alpha^{(t)}\Xi_k^{(t)} \end{aligned} \tag{20}$$

which is a linear system of $\Delta_k^{(t)}$ driven by noise ξ_k.

Now we focus on the steady-state covariance of $\Delta_t^{(t)}$,

$$S_k := \lim_{t\to\infty} \mathbb{E}[{\Delta_k^{(t)}}^\top \Delta_k^{(t)}] \tag{21}$$

which satisfies the discrete Lyapunov equation

$$S_k = A_k S_k A_k^\top + [\alpha^{(t)}]^2 \Sigma_k \tag{22}$$

where $A_k = I - \gamma^{(t)}L - \alpha^{(t)}\lambda_k I$ and Σ_k is the covariance matrix of the projected noise.

We denote the eigenvalues of W by $\{\lambda_1^W, \cdots, \lambda_n^W\}$. The system is stable only if the eigenvalues of A_k are all in $(-1, 1)$, from which we can derive the condition on $\alpha^{(t)}$ by

$$-1 < 1 - \gamma^{(t)} + \gamma^{(t)}\lambda_j^W - \alpha^{(t)}\lambda_k < 1 \quad \forall j \in [n] \tag{23}$$

By using the fact that W is a non-negative and doubly stochastic matrix which defines a communication topology, we have $\lambda_j^W \in (-1, 1]$ for all $j \in [n]$. Recall that x^* is a local minima of $F(x)$ that is locally strongly convex, we have $\lambda_k > 0$. Then we can derive the condition on $\alpha^{(t)}$ that makes the system stable, which is

$$\alpha^{(t)} < \frac{2 + (\lambda_{\min}^W - 1)\gamma^{(t)}}{\lambda_k} \tag{24}$$

From Eq. (24), we can derive that, for the same set of α and γ, the condition is easier to fulfill with smaller λ_k. Therefore, the system is stable, or the contraction of S_k happens earlier, with a smaller eigenvalue λ_k and with non-increasing step sizes.

With the contractions happening on the subspace spanned by eigenvectors with small eigenvalues, the consensus errors align with the eigenvectors that have larger eigenvalues, which is the dominant subspace of the Hessian.

A.3 Experiment details

A.3.1 Image Classification Experiments on CIFAR10

The selection of hyperparameters follows the original paper (Zagoruyko & Komodakis, 2016), and our baseline implementation perfectly matches its performance.

Category	Setting
General	
Number of epochs	200
Global batch size	128
Learning rate scheduler	Linear warm-up to 0.1 in the first 10 epochs, followed by cosine annealing until the end
Base optimizer	SGD with momentum (0.9), weight decay = 5×10^{-4}
Data shuffle	Randomly shuffled and split into N local datasets each epoch
Decentralized training	
Number of workers	8
Communication topology	One-peer ring (alternating between neighbors $i-1$ and $i+1$ across iterations)
DSGD-AC parameters	Exponent $p = 3$ (tuned based on experiments); $\gamma = 1$ during warm-up
BatchNorm calibration	Similar to the case in (Defazio et al., 2024), a calibration on the BatchNorm statistics is needed because there is a mismatch between the local models and the global average. To calibrate mismatched statistics, a full pass over the training set is conducted before validation. Only one calibration should be done if intermediate checkpoints are not evaluated.

A.3.2 Transformer on WMT14

The selection of hyperparameters follows the original paper (Vaswani et al., 2017), and our baseline implementation perfectly matches its performance.

Category	Setting
General	
Number of epochs	20
Global batch size	$\sim$25k tokens
Learning rate scheduler	Linear warm-up to 5×10^{-4} over the first 4000 iterations, then decay as $5 \times 10^{-4} \cdot (4000/t)^{0.5}$ (t = iteration index)
Base optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98$)
Data shuffle	Randomly shuffled and split into N local datasets each epoch
Decentralized training	
Number of workers	8
Communication topology	One-peer ring (alternating between neighbors $i-1$ and $i+1$ across iterations)
DSGD-AC parameters	Exponent $p = 3$ (tuned based on experiments); $\gamma = 1$ during warm-up
Normalization	Only layer normalization is used; no calibration needed

A.4 EXPERIMENT RESULTS

A.4.1 IMAGE CLASSIFICATION WITH WIDE RESNET

The complete statistics of the image classification task are deferred to this section due to the space limit. Even though comparing DSGD-AC with SAM-like methods is not our emphasis, we implement SAM (Foret et al., 2020) and the average-direction SAM (Bisla et al., 2022) and report their results for reference. We follow Foret et al. (2020) to use $\rho = 0.05$ in all the experiments, and use the same schedule of the variance of the random perturbations as described in the official GitHub repository[1] (Bisla et al., 2022).

Figures 5, 6, 7, and 8 and Table 3 present all the results on the image classification task. We summary the results as

- SAM always outperforms other methods at the cost of $2\times$ computation.
- DSGD-AC always achieves the best test loss among the methods with $1\times$ computation.
- AD-SAM outperforms DSGD-AC in the solution flatness only on experiments with WRN16-8, which is relatively smaller than WRN28-10.

Note that (1) for training loss, it is evaluated on the workers for decentralized training, and evaluated on perturbed points for SAM, (2) for test loss, it's evaluated on the global average model for decentralized training, (3) each curve for each algorithm is based on 3 runs with the same set of random seeds, and (4) the shaded parts correspond to the 95% confidence interval.

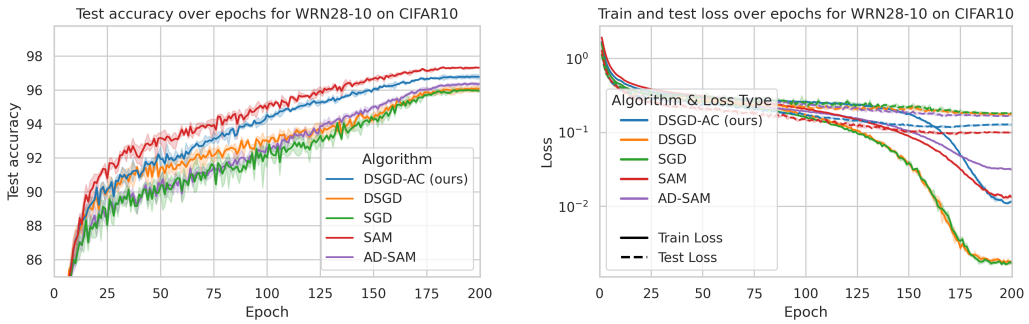

Figure 5: WRN28-10 on CIFAR-10. **Left**: Test accuracy on test set. For decentralized training, the accuracy is evaluated on the global average model. **Right**: Training and test losses.

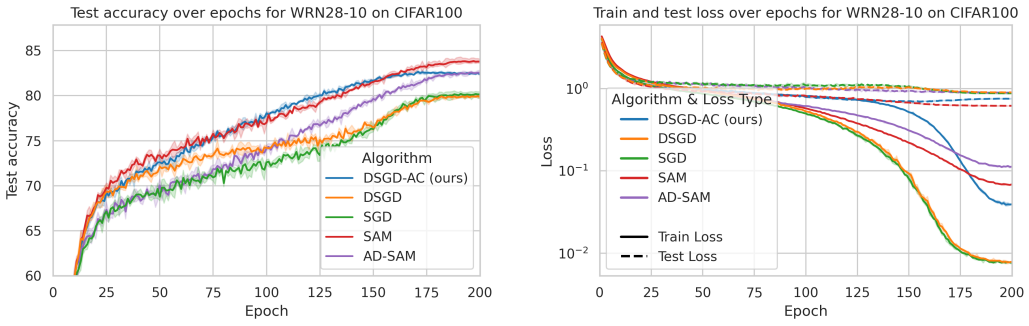

Figure 6: WRN28-10 on CIFAR-100. **Left**: Test accuracy on test set. For decentralized training, the accuracy is evaluated on the global average model. **Right**: Training and test losses.

[1] https://github.com/devansh20la/LPF-SGD/blob/master/codes/README.md

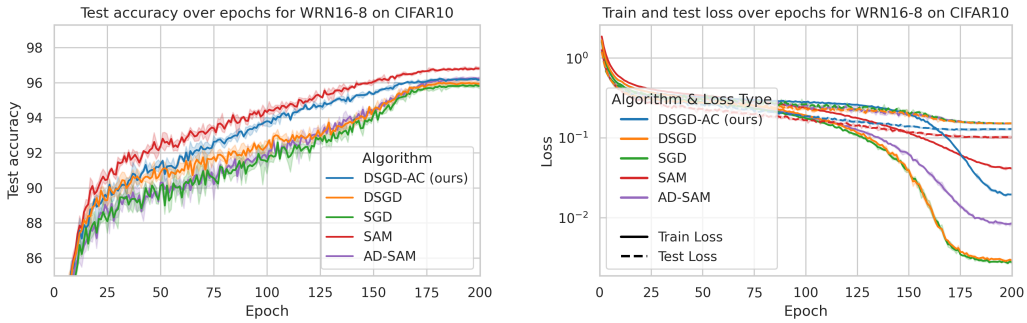

Figure 7: WRN16-8 on CIFAR-10. **Left**: Test accuracy on test set. For decentralized training, the accuracy is evaluated on the global average model. **Right**: Training and test losses.

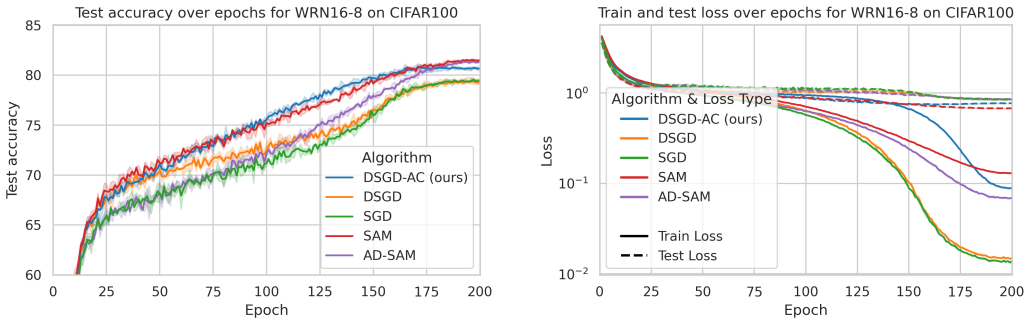

Figure 8: WRN16-8 on CIFAR-100. **Left**: Test accuracy on test set. For decentralized training, the accuracy is evaluated on the global average model. **Right**: Training and test losses.

Model	Dataset	Algorithm	Test Acc. (%) ↑	Test Loss ↓	Mean Top-1 Eigenvalue ↓	Computation ↓
WRN28-10	CIFAR-10	DSGD	96.07 ± 0.13	0.176 ± 0.005	22.4360 ± 3.9916	1x
		SGD	95.96 ± 0.14	0.182 ± 0.004	16.8485 ± 0.3251	1x
		DSGD-AC	96.77 ± 0.11	0.128 ± 0.003	8.9693 ± 0.3514	1x
		AD-SAM	96.37 ± 0.11	0.168 ± 0.002	24.9059 ± 1.6212	1x
		SAM	**97.33 ± 0.04**	**0.100 ± 0.002**	**0.3523 ± 0.0312**	2x
	CIFAR-100	DSGD	79.86 ± 0.22	0.899 ± 0.008	49.5719 ± 4.8022	1x
		SGD	80.15 ± 0.42	0.878 ± 0.020	37.3799 ± 2.8886	1x
		DSGD-AC	82.38 ± 0.09	0.755 ± 0.008	19.8061 ± 0.6653	1x
		AD-SAM	82.57 ± 0.31	0.891 ± 0.007	32.6371 ± 2.3362	1x
		SAM	**83.79 ± 0.25**	**0.618 ± 0.003**	**1.7295 ± 0.0385**	2x
WRN16-8	CIFAR-10	DSGD	95.94 ± 0.11	0.152 ± 0.001	18.1998 ± 0.6427	1x
		SGD	95.81 ± 0.13	0.153 ± 0.003	17.4934 ± 1.6191	1x
		DSGD-AC	96.17 ± 0.04	0.129 ± 0.003	11.8250 ± 0.4883	1x
		AD-SAM	96.25 ± 0.12	0.152 ± 0.002	8.5178 ± 0.5453	1x
		SAM	**96.81 ± 0.08**	**0.102 ± 0.003**	**1.3928 ± 0.0586**	2x
	CIFAR-100	DSGD	79.25 ± 0.26	0.854 ± 0.016	36.1998 ± 3.8028	1x
		SGD	79.42 ± 0.18	0.849 ± 0.015	33.7733 ± 0.7897	1x
		DSGD-AC	80.67 ± 0.11	0.771 ± 0.005	19.8032 ± 0.1652	1x
		AD-SAM	81.36 ± 0.06	0.858 ± 0.004	17.5450 ± 1.2583	1x
		SAM	**81.51 ± 0.08**	**0.677 ± 0.003**	**4.7932 ± 0.1957**	2x

Table 3: Algorithm comparison on image classification including SAM (Foret et al., 2020) and average-direction SAM (Bisla et al., 2022).

A.4.2 SENSITIVITY ANALYSIS OF THE HYPERPARAMETER IN DSGD-AC

In all experiments, we use $p = 3$ for DSGD-AC, which is based on experiment tuning. The test results with $p = \{0, 1, 2, 3, 4, 5\}$ are presented in Figure 9 and Table 4. The tracked average norm of consensus errors with varying p is shown in Figure 10.

Note that DSGD-AC with $p = 0$ is equivalent to DSGD. The results demonstrate the effectiveness of introducing p and DSGD-AC, and $p = 3$ brings the best performance.

Figure 9: DSGD(-AC) on WRN28-10 on CIFAR-10 with varying p. **Left**: Test accuracy on test set. For decentralized training, the accuracy is evaluated on the global average model. **Right**: Training and test losses.

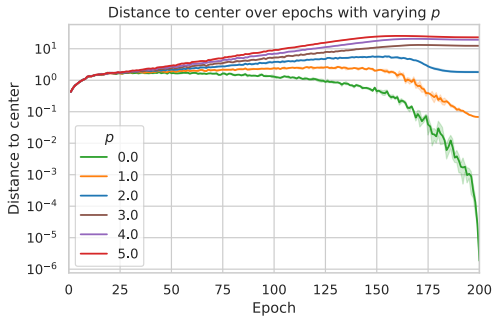

Figure 10: Average norm of consensus errors over epochs with varying p.

p	Test Accuracy (%) ↑	Train Loss ↓	Test Loss ↓
0	96.07 ± 0.13	$\mathbf{0.002} \pm 0.000$	0.176 ± 0.005
1	96.26 ± 0.14	$\mathbf{0.002} \pm 0.000$	0.159 ± 0.003
2	$\underline{96.58} \pm 0.18$	0.003 ± 0.000	0.141 ± 0.006
3	$\mathbf{96.77} \pm 0.11$	0.012 ± 0.000	$\underline{0.128} \pm 0.003$
4	$\underline{96.53} \pm 0.13$	0.024 ± 0.001	$\mathbf{0.127} \pm 0.004$
5	96.37 ± 0.04	0.032 ± 0.001	0.130 ± 0.002

Table 4: Sensitivity analysis of parameter p in the WRN28-10 on CIFAR10 experiment. The best value is **bold**, and the second best is underlined.

A.5 USE OF LARGE LANGUAGE MODELS

During the development of the paper, we used LLMs to polish the text without changing its original meaning.