Decentralized SGD with Controlled Disagreement Finds Flatter Minima

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. This work challenges this view by introducing decentralized SGD with Adaptive Consensus (DSGD-AC), which uses a time-dependent scaling mechanism to maintain consensus errors throughout the training. We show that adaptive consensus changes the stationary variance of disagreement modes by balancing two effects: it preserves consensus-error magnitude through weaker graph damping while still allowing curvature-dependent damping to shape the disagreement directions. This balance can produce a stronger Hessian-weighted loss-envelope penalty around the deployed model, even when normalized Hessian alignment is weaker than in standard DSGD. Empirical results on image classification show that DSGD-AC reaches flatter solutions and higher test accuracy than standard DSGD and even centralized SGD. Together, these results support 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 training lets workers exchange model parameters only with neighbors, avoiding the cost of global synchronization and all-reduce communication (Abadi et al., 2016; Li et al., 2020). This reduces latency and eliminates single points of failure, making decentralized approaches attractive for geographically distributed systems (Dhasade et al., 2023; Gholami & Seferoglu, 2024) and GPU clusters (Lian et al., 2017; Assran et al., 2019; Wang et al., 2025).

Despite its practical appeal, decentralized training methods such as the decentralized stochastic gradient descent (DSGD) are widely regarded as inferior to centralized training in terms of convergence and generalization, even when data distributions of workers are i.i.d. This gap was largely attributed to consensus errors, the persistent discrepancies in the model parameters between different workers (Alghunaim & Yuan, 2022; Zhu et al., 2022). Prior work has therefore focused on reducing these errors through improved 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 make decentralized training closer to its centralized counterpart.

However, this perspective neglects the potential constructive role of consensus errors. Rather than acting as detrimental noise, these discrepancies may serve as structured perturbations that encourage preference for flatter minima in the loss landscape, solutions that strongly correlate with better generalization (Jiang et al., 2019). This view draws inspiration from sharpness-aware minimization strategies (Foret et al., 2020; Bisla et al., 2022; Li et al., 2024b), which improve model robustness and generalization by explicitly introducing curvature-aware perturbations.

In this study, we challenge the conventional view by introducing Decentralized SGD with Adaptive Consensus (DSGD-AC), an algorithm that strategically maintains consensus errors through an adaptive, time-dependent scaling mechanism. Our theory characterizes the stationary variance of disagreement modes and shows how the adaptive consensus factor balances the magnitude and curvature profile of consensus errors, inducing a stronger Hessian-weighted penalty around the deployed global average. We verify the theory with comprehensive empirical results and diagnostics on Hessian information. DSGD-AC incurs negligible additional

computational cost relative to standard SGD or DSGD, while retaining the communication efficiency and runtime advantages of decentralized training.

The main contributions of this work are threefold. First, we propose DSGD-AC, an adaptive consensus algorithm that maintains theoretically motivated consensus errors, finds flatter minima, and improves generalization on deep learning tasks at minimal computational cost. Second, we provide theoretical analysis and empirical diagnostics of the modal variance and consensus error radius, showing that the controlled errors induce a stronger Hessian-weighted penalty and are associated with flatter solutions. Third, we evaluate DSGD-AC against strong baselines on deep learning benchmarks, verifying its practical effectiveness.

1.1 Related work

Canonical view of consensus errors The prevailing view in decentralized training is that models should minimize the consensus errors and approximate 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 constructive role of consensus errors by showing that controlled disagreement can induce a meaningful Hessian-weighted loss-envelope penalty, and we propose DSGD-AC as a practical way to exploit this effect in deep learning tasks.

Explorations beyond the canonical view Since the canonical perspective dominates, research into the potential benefits of consensus errors remains sparse. Kong et al. (2021) empirically identified error thresholds and noted advantages in specific training phases; however, they did not explore regimes where errors exceed those of ring-topology DSGD, and their control scheme showed limited gains. 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 how the magnitude and Hessian-weighted effect of consensus errors are shaped by gradient noise and curvature-dependent damping, and, by proposing DSGD-AC, shows how decentralized training can use disagreement constructively beyond the large-batch setting.

Explicit curvature-related perturbations If the global average is taken as the deployed model (Zhu et al., 2023), decentralized learning can be interpreted as training the average model, and worker deviations act as perturbations around it. 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 improve it further. However, to achieve the best performance, these algorithms typically require additional gradient evaluations and increase the computational cost significantly. Our work uses consensus errors as free perturbations that can enhance generalization without introducing extra computation or memory than DSGD.

2 Problem setting and notation

We consider decentralized training in a data center where the full dataset is accessible to all workers. The training uses a standard distributed data sampler: at the beginning of each epoch, the dataset is reshuffled and evenly partitioned across workers, yielding i.i.d. data distributions in expectation. This setting corresponds to the standard decentralized optimization regime studied by Assran et al. (2019); Ying et al. (2021); Kong et al. (2021); Zhu et al. (2023); Wang et al. (2025).

Decentralized optimization We consider n workers collaborating to train a d-dimensional model. Let $[n]$ denote the set of integers $\{1, 2, \ldots, n\}$. Each worker $i \in [n]$ holds a local objective determined by its local

dataset $\mathcal{D}_i$:

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

Under the i.i.d. setting considered here, all workers share the same underlying objective but operate on different mini-batches drawn from the shared dataset. Each worker maintains its own local model x_i. The workers collaboratively solve

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

where the constraint enforces consensus across workers.

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

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

where $\mathcal{N}(i)$ is the neighbor set of worker i (including itself), W is a symmetric, non-negative, doubly stochastic matrix defining with $W_{ij} = 0$ if $i \notin \mathcal{N}_j$), and $s_i^{(t)}$ is the mini-batch sampled by worker i at iteration t.

Following standard notation, we denote the global average by $\bar{x}^{(t)} := \frac{1}{n}\sum_{i=1}^n x_i^{(t)}$, the consensus errors $\delta_i^{(t)} := x_i^{(t)} - \bar{x}^{(t)}$ and the matrix forms $X^{(t)} := [x_1^{(t)},\cdots,x_n^{(t)}]^\top$, $G^{(t)} := [g_1^{(t)}, g_2^{(t)},\cdots,g_n^{(t)}]^\top$, $\Delta^{(t)} = [\delta_1^{(t)},\cdots,\delta_n^{(t)}]^\top$, and $\bar{X}^{(t)} = [\bar{x}^{(t)},\cdots,\bar{x}^{(t)}]^\top$. We also denote the center projector $P = I - \frac{1}{n}11^\top$ and the graph Laplacian $L = I - W$. Then, the DSGD update in Eq. (3) can be written in matrix form

$$X^{(t)} = X^{(t-1)} - \alpha^{(t)}G^{(t)} + (W - I)X^{(t-1)} \tag{4}$$

We focus on the adapt-while-communication variant, which has been shown to be more runtime efficient (Lian et al., 2017; Assran et al., 2019; Wang et al., 2025) and to share the same generalization bound as the adapt-then-communication variant (Bellet et al., 2023).

2.1 Assumptions

Our theoretical analysis relies on the following assumptions.

Assumption 1 (Twice-differentiable objectives)**.** *The objective functions f_i are at least twice differentiable.*

Assumption 2 (Decentralized communication topology)**.** *The graph is connected and undirected. W is non-negative, symmetric, and doubly stochastic.*

Assumption 3 (I.i.d. data distributions)**.** *All workers have access to the full dataset, so that*

$$f_1 = f_2 = \cdots = f_n = f$$

Assumption 4 (Martingale-difference gradient noise)**.** *Let $g_i^{(t)}$ be the mini-batch gradient on worker i. Under $f_i = f$, its first-order expansion around $\bar{x}^{(t-1)}$ is*

$$g_i^{(t)} = \nabla f(\bar{x}^{(t-1)}) + H^{(t)}\delta_i^{(t-1)} + r_i^{(t)} + \xi_i^{(t)},$$

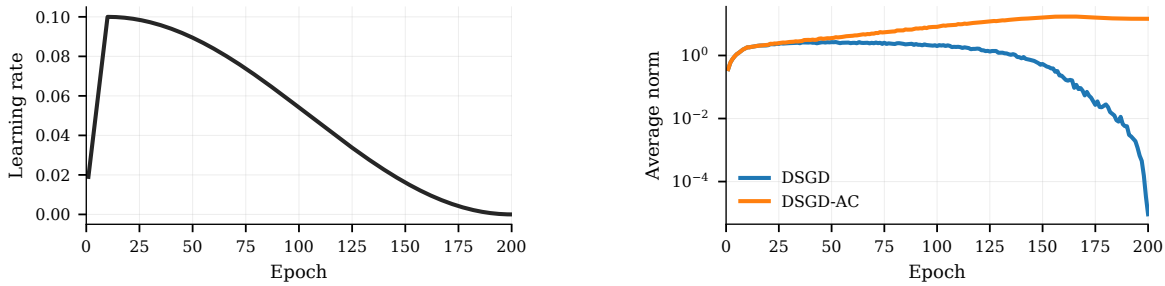
where $H^{(t)} = \nabla^2 f(\bar{x}^{(t-1)})$ is the Hessian, $r_i^{(t)}$ the Taylor residual, and $\xi_i^{(t)}$ mini-batch noise. In matrix form,

$$G^{(t)} = \mathbf{1}\nabla f(\bar{x}^{(t-1)})^\top + \Delta^{(t-1)}H^{(t)} + R^{(t)} + \Xi^{(t)}.$$

Let $\mathcal{F}_{t-1}$ be the information available before sampling the mini-batches at iteration t. The noise is conditionally mean zero:

$$\mathbb{E}\left[\Xi^{(t)} \mid \mathcal{F}_{t-1}\right] = 0$$

Assumption 5 (Frozen Hessian local window)**.** *For the modal analysis, we study the local linearized dynamics in a short late-training window, setting $H^{(t)} = H_f$ and dropping the Taylor residual $R^{(t)}$. All modal variance and radius statements below refer to this local model.*



(a) Cosine annealing learning rate schedule

(b) Norm of consensus errors evaluated across epochs

Figure 1: Decentralized training of WRN16-8 on CIFAR-100 with 8 workers and the one-peer ring topology.

3 DSGD-AC and motivating observations

This section motivates, develops, and analyzes DSGD-AC in three steps. We first show that consensus errors in DSGD vanish as the learning rate decays and argue that this limits its potential in generalization. We then propose DSGD-AC as a remedy and show empirically that it reaches flatter minima and higher test accuracy. Finally, we provide a theoretical analysis of the consensus-error dynamics, characterizing how the adaptive factor controls the magnitude and curvature profile of the maintained errors. All empirical claims are supported by training WRN16-8 (Zagoruyko & Komodakis, 2016) on CIFAR-100 (Krizhevsky et al., 2009) with 8 workers in a one-peer ring topology under cosine annealing with linear warm-up (see Figure 1a). Additional hyperparameter details are given in Appendix A.3.

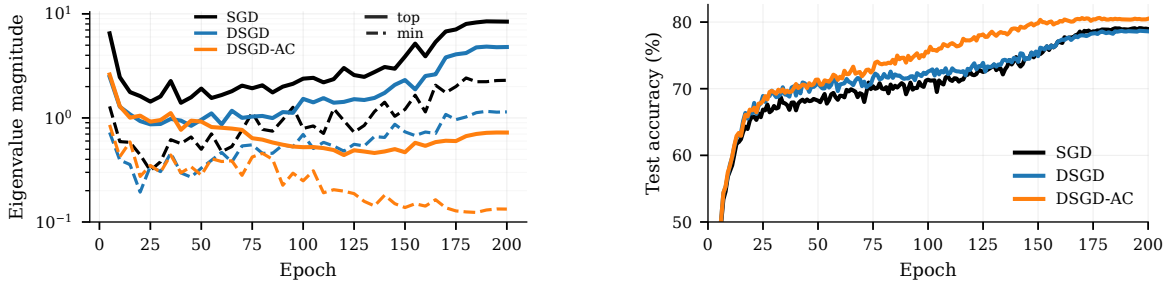
3.1 Vanishing consensus error in DSGD

The conventional view of consensus errors as a nuisance was challenged by Zhu et al. (2023), who argued that one should view worker deviations $\delta_i^{(t)}$ as random perturbations around the global average $\bar{x}^{(t)}$, and proved that DSGD is asymptotically equivalent to average-direction SAM (Bisla et al., 2022). Under this interpretation, consensus errors provide implicit sharpness-aware regularization at no additional cost. However, the practical benefit of this observation is limited: unlike SAM, which maintains a fixed perturbation radius ρ, the consensus-error radius $\frac{1}{n}\sum_{i=1}^{n}\|\delta_i^{(t)}\|$ vanishes asymptotically as the learning rate decays (Lian et al., 2017; Zhu et al., 2022). The implicit regularization thus disappears precisely when it is most needed, in the final stages of training where the model converges. This is confirmed empirically in Figure 1b.

To see why the consensus-error radius vanishes, it is instructive to view DSGD as stochastic gradient descent on the per-step surrogate $J^{(t)}$, defined as

$$J^{(t)}(x_1,\cdots,x_n) = \underbrace{\sum_{i=1}^{n} f_i(\bar{x}^{(t)})}_{\text{objective on deployed model}} + \underbrace{\sum_{i=1}^{n}[f_i(x_i^{(t)}) - f_i(\bar{x}^{(t)})]}_{\text{sharpness}} + \underbrace{\frac{1}{4\alpha^{(t)}}\sum_{i,j\in[n]} W_{ij}\|x_i^{(t)} - x_j^{(t)}\|^2}_{\text{consensus regularizer}} \tag{5}$$

Differentiating $J^{(t)}$ with respect to x_i and substituting back into the gradient descent update recovers the DSGD update in Eq. (3) exactly. The sharpness penalty measures the curvature seen by the workers: when the local models scatter away from $\bar{x}^{(t)}$, this term grows and penalizes sharp regions of the loss. As $\alpha^{(t)} \to 0$, however, the consensus regularizer weight $1/(4\alpha^{(t)})$ diverges and dominates $J^{(t)}$, forcing $x_i^{(t)} \to \bar{x}^{(t)}$ for all i. The sharpness penalty vanishes with the consensus errors, and the only remaining term is $\sum_i f_i(\bar{x}^{(t)})$, the same objective minimized by synchronous SGD. Maintaining a non-vanishing consensus-error radius throughout training is therefore necessary to preserve the potential regularization effect.



(a) Hessian eigenvalue magnitudes measured by the Lanczos algorithm with 30 iterations

(b) Test accuracy evaluated at the deployed models (global averaged center for decentralized training)

Figure 2: Training performance and curvature for WRN16-8 on CIFAR-100 using 8 workers and the one-peer ring topology. DSGD-AC improves test accuracy and reaches a flatter local region than DSGD and synchronous SGD.

3.2 DSGD with adaptive consensus

To maintain a non-vanishing perturbation radius throughout training, we propose DSGD with adaptive consensus (DSGD-AC), detailed in Algorithm 1. The only modification to DSGD is a time-varying scalar factor $\gamma^{(t)} \in [0, 1]$ that scales the consensus step at each iteration. In terms of the surrogate $J^{(t)}$ in Eq. (5), this changes the consensus-regularizer weight from $1/(4\alpha^{(t)})$ to $\gamma^{(t)}/(4\alpha^{(t)})$, preventing it from dominating as $\alpha^{(t)} \to 0$. The modified lines are highlighted in Algorithm 1, and the algorithm takes the global average $\bar{x}^{(t)}$ 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 epochs (E), the number of batches per epoch (T), initialization ($x^{(0)}$), 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**

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

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

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

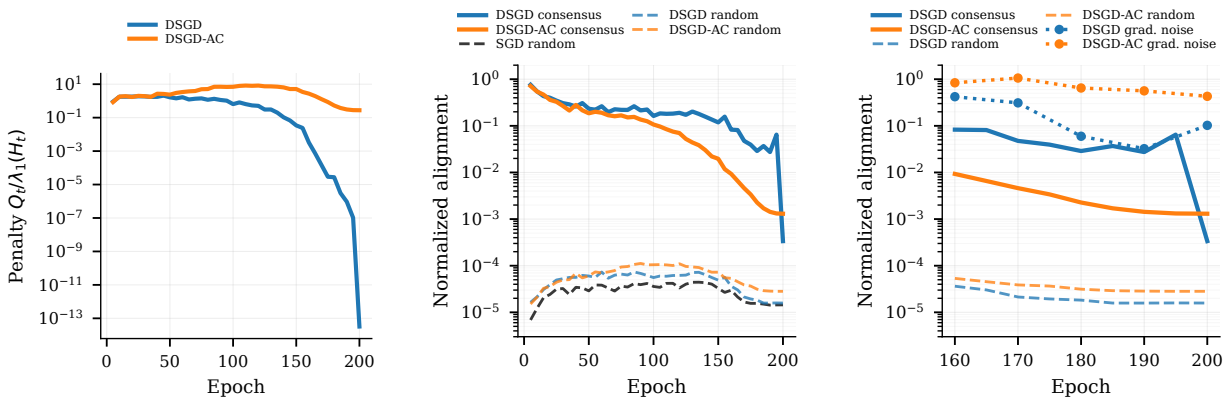
end

Note that $\alpha^{(t)}$ is determined by a learning rate scheduler such as cosine annealing (Loshchilov & Hutter, 2016), and $\alpha_{\max}$ is the maximal learning rate throughout the training, which ensures $\gamma^{(t)}$ is in the range $[0, 1]$. The schedule $\gamma^{(t)} = [\alpha^{(t)}/\alpha_{\max}]^p$ ties the consensus factor directly to the learning rate, so the two decay together. We use $p = 3$ in the showcase experiments, which is chosen based on hyperparameter tuning.

As shown in Figure 1b, DSGD-AC keeps the consensus errors non-vanishing after the learning-rate decay, while also reaching a smaller top Hessian eigenvalue, a narrower Hessian spectrum, and improved test accuracy over DSGD and even synchronous SGD (Figures 2a and 2b). The experiments in Section 4 show that these gains persist across additional models, worker counts, and communication topologies.

3.3 Empirical Hessian diagnostics

We next investigate why maintaining larger disagreement does not simply degrade training. The loss envelope around the deployed average model $\bar{x}^{(t)}$ separates the effect of the worker deviations $\delta_i^{(t)}$ from the objective



(a) Curvature exposure Q_t. (b) Alignment A_t and baseline B_t. (c) Gradient-noise alignment.

Figure 3: Consensus-error Hessian diagnostics for WRN16-8 on CIFAR-100: (a) normalized and unnormalized penalties, (b) consensus and random-direction alignment, and (c) late-training gradient-noise alignment. Except for $\mathcal{E}_t$, all quantities are normalized by the top Hessian eigenvalue.

value at the average. A second-order expansion gives

$$\sum_{i=1}^{n} f(\bar{x}^{(t)} + \delta_i^{(t)}) \approx n f(\bar{x}^{(t)}) + \frac{1}{2} \sum_{i=1}^{n} (\delta_i^{(t)})^\top H^{(t)} \delta_i^{(t)}, \tag{6}$$

where the first-order terms cancel because $\sum_i \delta_i^{(t)} = 0$, and the remainder is negligible when the consensus-error radius is small relative to the curvature scale of f. The quadratic term is a Hessian-weighted penalty on the consensus errors, concentrated on positive-curvature directions.

To assess this effect empirically, we track three quantities defined in terms of the Hessian $H^{(t)} := \nabla^2 f(\bar{x}^{(t)})$, its largest eigenvalue $\lambda_1(H^{(t)})$, and the consensus errors $\Delta^{(t)}$. The first diagnostic is the curvature exposure, the Hessian-weighted consensus-error penalty from the loss envelope in Eq. (6):

$$Q_t := \frac{\operatorname{tr}\big(\Delta^{(t)} H^{(t)} (\Delta^{(t)})^\top\big)}{\lambda_1(H^{(t)})} = \sum_{i=1}^{n} \frac{(\delta_i^{(t)})^\top H^{(t)} \delta_i^{(t)}}{\lambda_1(H^{(t)})}. \tag{7}$$

The second is the normalized alignment score, separating the directional component of Q_t from its radius:

$$A_t := \frac{\operatorname{tr}\Big(\widetilde{\Delta}^{(t)} H^{(t)} (\widetilde{\Delta}^{(t)})^\top\Big)}{\lambda_1(H^{(t)})}, \tag{8}$$

Here, $\widetilde{\Delta}^{(t)} := \Delta^{(t)} / \|\Delta^{(t)}\|_F$ so $Q_t = \|\Delta^{(t)}\|_F^2 \cdot A_t$, decomposing the penalty into radius and alignment. The third is the random-direction baseline, the expected normalized alignment of Rademacher directions,

$$B_t := \frac{\mathbb{E}_u\big[u^\top H^{(t)} u / \|u\|^2\big]}{\lambda_1(H^{(t)})} = \frac{\operatorname{tr}(H^{(t)})}{d \cdot \lambda_1(H^{(t)})}, \tag{9}$$

estimated using 50 Hutchinson trace probes. Together, Q_t, A_t, and B_t diagnose how the consensus errors interact with the local curvature at the deployed model; all three are normalized by $\lambda_1(H^{(t)})$ to remove loss-scaling bias and facilitate comparison across runs. Further evaluation details are in Appendix A.3.3.

The three-way comparison in Figure 3 is instructive. In standard DSGD, the consensus-error radius collapses as $\alpha^{(t)} \to 0$, and Q_t collapses with it despite maintained alignment A_t. B_t shows that random directions have poor Hessian alignment, so their curvature exposure is low. DSGD-AC occupies a different position: its normalized alignment A_t is no stronger than DSGD, but, by preventing radius collapse, it maintains a

substantially larger Q_t than either alternative. The preference of DSGD-AC for flatter minima is therefore better understood as preservation of curvature exposure than as improvement of Hessian alignment.

This rules out two simpler explanations. DSGD-AC does not find flatter minima by aligning its consensus direction more strongly with the top Hessian eigenspace than DSGD. Nor do the consensus errors act as random noise: Figure 3b shows that A_t remains well above the random baseline B_t for both methods throughout late training. The maintained above-random alignment is instead consistent with the anisotropic structure of SGD noise in deep networks, as examined in the following section.

3.4 Modal explanation of controlled disagreement

The empirical diagnostics in Section 3.3 show that DSGD-AC preserves curvature exposure by maintaining a non-vanishing consensus-error radius. We now explain the mechanism theoretically. The central message is simple: decreasing γ selectively preserves disagreement in low-curvature directions, while Hessian damping continues to suppress sharp directions. The analysis proceeds in three steps. Lemma 1 diagonalizes the disagreement dynamics into independent graph-Hessian modes. Theorem 1 gives each mode's stationary variance. Corollaries 1 and 2 translate these into radius predictions for $\gamma = \alpha^p$. Proofs are in Appendix A.1.

We start from the matrix form of DSGD-AC, analogous to Eq. (4):

$$\begin{aligned} X^{(t)} &= X^{(t-1)} - \alpha^{(t)} G^{(t)} + \gamma^{(t)}(W - I)X^{(t-1)} \\ &= (I - \gamma^{(t)} L)X^{(t-1)} - \alpha^{(t)} G^{(t)} \end{aligned} \tag{10}$$

Multiplying by the center projector P removes the average model and isolates the disagreement $\Delta^{(t)} = PX^{(t)}$. Since W is doubly stochastic under Assumption 2, $PL = LP$ and we find the disagreement recursion

$$\Delta^{(t)} = P(I - \gamma^{(t)} L)X^{(t-1)} - \alpha^{(t)} PG^{(t)} = (I - \gamma^{(t)} L)\Delta^{(t-1)} - \alpha^{(t)} PG^{(t)} \tag{11}$$

Equation (11) already shows the two competing forces: mixing damps disagreement through $\gamma^{(t)} L$, while the worker-wise stochastic gradients continually inject disagreement through $PG^{(t)}$. To see which Hessian directions receive this injected energy, we linearize the gradients around the deployed average model and diagonalize both the graph Laplacian and the Hessian.

Lemma 1 (Linearized modal dynamics)**.** *Under Assumptions 1–5, the local linearized dynamics in a late-training window decouple exactly into independent scalar recursions. Let $L = V \operatorname{diag}(\mu_1, \dots, \mu_n) V^\top$ with $2 \geq \mu_1 \geq \cdots \geq \mu_{n-1} > \mu_n = 0$, and let $H_f = U \operatorname{diag}(\lambda_1, \dots, \lambda_d) U^\top$ with $\lambda_1 \geq \cdots \geq \lambda_d$. Define $z_{j,k}^{(t)} = v_j^\top \Delta^{(t)} u_k$ and $\xi_{j,k}^{(t)} = v_j^\top \Xi^{(t)} u_k$. Then each graph-Hessian mode satisfies*

$$z_{j,k}^{(t)} = (1 - \gamma^{(t)}\mu_j - \alpha^{(t)}\lambda_k) z_{j,k}^{(t-1)} - \alpha^{(t)} \xi_{j,k}^{(t)}. \tag{12}$$

The scalar recursion in Lemma 1 is the basic mechanism behind DSGD-AC, and explains many of the empirical observations in Section 3.3. The effective damping of mode (j, k) is $d_{j,k} := \gamma\mu_j + \alpha\lambda_k$, the sum of a graph term and a curvature term. Decreasing γ weakens only the graph damping, while the positive Hessian curvature continues to suppress sharp directions independently. Theorem 1 makes this precise by computing the stationary variance of each mode as a function of $d_{j,k}$, the noise variance $\sigma_{j,k}^2$, and the learning rate α.

Theorem 1 (Stationary modal variance)**.** *In the setting of Lemma 1, fix constant α and γ. Assume the conditional modal noise variance is constant in the local window:*

$$\sigma_{j,k}^2 := \mathbb{E}\left[\left(\xi_{j,k}^{(t)}\right)^2 \mid \mathcal{F}_{t-1}\right].$$

Assume the stability condition $0 < d_{j,k} < 2$. When $d_{j,k} \leq 0$, the mode is unstable, and the variance diverges; this can occur for negative-curvature directions when $\alpha|\lambda_k| > \gamma\mu_j$. Under stability, for every disagreement mode $j \in \{1, \dots, n-1\}$, the stationary modal variance is

$$\mathbb{E}[z_{j,k}^2] = \frac{\alpha^2 \sigma_{j,k}^2}{d_{j,k}(2 - d_{j,k})}. \tag{13}$$

Theorem 1 gives the exact stationary variance for fixed α and γ. Remark 1 derives order-of-magnitude bounds that are more useful in practice, separating two regimes: modes where graph damping dominates (low curvature) and modes where Hessian damping dominates (high curvature). These bounds are the key tool for understanding how γ shapes the radius.

Remark 1 (Modal scaling bounds)**.** *Suppose the conditions of Theorem 1 hold and $d_{j,k} \leq d_{\max} < 2$. Then*

$$\frac{\alpha^2 \sigma_{j,k}^2}{2d_{j,k}} \leq \mathbb{E}[z_{j,k}^2] \leq \frac{\alpha^2 \sigma_{j,k}^2}{(2 - d_{\max}) d_{j,k}}. \tag{14}$$

If a mode is mixing-dominated, meaning $|\alpha\lambda_k| \leq \rho\gamma\mu_j$ for a fixed $\rho \in [0, 1)$, then

$$\mathbb{E}[z_{j,k}^2] = \Theta\left(\frac{\alpha^2 \sigma_{j,k}^2}{\gamma\mu_j}\right). \tag{15}$$

If $\lambda_k > 0$, then

$$\mathbb{E}[z_{j,k}^2] = \Theta\left(\frac{\alpha^2 \sigma_{j,k}^2}{\gamma\mu_j + \alpha\lambda_k}\right), \tag{16}$$

so positive high-curvature directions are damped by the Hessian term. For negative-curvature directions, the same formula holds whenever $0 < d_{j,k} < 2$, but the Hessian term reduces $d_{j,k}$ and can increase the variance.

The remark shows that low-curvature modes scale as $\Theta(\alpha^2/\gamma)$ while high-curvature modes are suppressed to $\mathcal{O}(\alpha)$. Corollary 1 sums these contributions across modes to characterize the total disagreement radius, and identifies the condition under which the low-curvature contribution dominates.

Corollary 1 (Conditional radius scaling from low-curvature modes)**.** *Let $\mathcal{S}_{\text{low}} = \{(j,k) : 1 \leq j \leq n - 1,\ |\alpha\lambda_k| \leq \rho\gamma\mu_j\}$ for some fixed $\rho \in [0, 1)$, and let $\mathcal{S}_{\text{high}}^+ = \{(j,k) : 1 \leq j \leq n - 1,\ \lambda_k \geq c > 0\}$ for a fixed curvature threshold c. The set $\mathcal{S}_{\text{low}}$ is non-empty only when genuinely flat directions exist, i.e., directions with $|\lambda_k| \ll \gamma\mu_j/\alpha$. Define the active low-curvature modal weight*

$$M_{\text{low}}(\alpha, \gamma) := \sum_{(j,k)\in\mathcal{S}_{\text{low}}} \frac{\sigma_{j,k}^2}{\mu_j}. \tag{17}$$

If $M_{\text{low}}(\alpha, \gamma) > 0$ and the modes in $\mathcal{S}_{\text{low}}$ are stable, then

$$\sum_{(j,k)\in\mathcal{S}_{\text{low}}} \mathbb{E}[z_{j,k}^2] = \Theta\left(\frac{\alpha^2}{\gamma} M_{\text{low}}(\alpha, \gamma)\right). \tag{18}$$

If $M_{\text{low}}(\alpha, \gamma)$ is bounded above and below by positive constants, this is $\Theta(\alpha^2/\gamma)$. If $\sigma_{j,k}^2 \leq \sigma_+^2 < \infty$ on $\mathcal{S}_{\text{high}}^+$ and the corresponding modes are stable, then

$$\sum_{(j,k)\in\mathcal{S}_{\text{high}}^+} \mathbb{E}[z_{j,k}^2] = O(\alpha). \tag{19}$$

The low-curvature contribution dominates when $\alpha^2/\gamma = \Omega(\alpha)$, i.e. when $\gamma = O(\alpha)$. For $\gamma = \alpha^p$, this holds for all $p \geq 1$; the practically relevant regime $p \geq 2$ is analyzed in Corollary 2.

The set $\mathcal{S}_{\text{low}}$ is non-empty whenever H_f has a nontrivial null space, since any almost-flat direction satisfies $|\alpha\lambda_k| \leq \rho\gamma\mu_j$. More generally, low-rank Hessian structure in neural networks motivates the presence of such modes (Keskar et al., 2016; Singh et al., 2021; Song et al., 2024; Ben Arous et al., 2024).

The radius scaling in Corollary 1 is stated for general γ. Corollary 2 specializes to the schedule $\gamma = \alpha^p$ used in DSGD-AC, which gives a direct prediction for how the maintained radius depends on p as α decays.

Corollary 2 (Conditional scaling under $\gamma = \alpha^p$)**.** *Set $\gamma = \alpha^p$ up to a fixed constant factor, with $p \geq 2$. For mixing-dominated modes, the active condition $|\lambda_k| \leq \rho\mu_j\alpha^{p-1}$ tightens as $\alpha \to 0$, so the following applies only to modes that remain active. For such modes,*

$$\mathbb{E}[z_{j,k}^2] = \Theta\left(\alpha^{2-p}\sigma_{j,k}^2/\mu_j\right). \tag{20}$$

For $p = 2$, the variance is of constant order in each active mode. For $p > 2$, it grows as $\alpha \to 0$; if unchecked, this can destabilize training, consistent with the performance degradation observed for $p > 4$ in Table 1.

These results show the mechanism behind DSGD-AC. Reducing γ preserves disagreement magnitude in active low-curvature modes, while the Hessian term $\alpha\lambda_k$ continues to suppress positive high-curvature modes. For $\gamma = \alpha^p$, $p = 2$ maintains constant-order variance in surviving mixing-dominated modes, whereas $p > 2$ increases their linearized stationary variance as α decays. Because the active condition tightens over training, the maintained radius is filtered toward flatter directions rather than spread uniformly across the spectrum.

The curvature exposure Q_t introduced in Section 3.3 can be written in modal coordinates as

$$Q_t = \sum_{i=1}^{n}(\delta_i^{(t)})^\top H_f \delta_i^{(t)}/\lambda_1(H_f) = \sum_{k=1}^{d}\lambda_k \sum_{j=1}^{n-1}(z_{j,k}^{(t)})^2/\lambda_1(H_f). \tag{21}$$

This decomposition separates the two effects identified empirically: the magnitude of the maintained disagreement, controlled by γ, and its distribution across Hessian eigen directions, controlled by the modal noise variances and curvature-dependent damping $d_{j,k}$. A method with weaker normalized Hessian alignment can still produce larger Q_t if its radius is sufficiently larger — precisely what DSGD-AC achieves.

Remark 2 (Source-noise anisotropy and Hessian alignment)**.** *The modal formula in Remark 1 implicitly explains why the above-random alignment $A_t > B_t$ observed in Figure 3b should not be attributed to adaptive consensus. If the injected noise $\sigma_{j,k}^2$ were isotropic in the Hessian eigen basis, the curvature term in $d_{j,k} = \gamma\mu_j + \alpha\lambda_k$ would damp high-curvature modes more strongly, and alignment would not exceed the random baseline. The observed alignment is instead consistent with anisotropic source noise: SGD noise in deep networks is known to concentrate energy in sharp directions (Wu et al., 2022; Ziyin et al., 2022), and Figure 3c verifies this directly for late-training gradient noise. Worker disagreements in both DSGD and DSGD-AC are driven by this structured noise, and graph mixing and curvature-dependent damping determine how much of each injected mode survives.*

4 Numerical Experiments

In this section, we present extensive numerical experiments on image classification using Wide ResNets (Zagoruyko & Komodakis, 2016) trained on CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009) under a variety of distributed setups. We follow the hyperparameter settings from the original papers and reproduce comparable baseline performance to ensure fair comparisons. All models are trained for 200 epochs. The global batch size is set to 128 for 8 workers and is linearly scaled with the number of workers; mixed-precision training is enabled by default. Since the networks use batch normalization layers and the training was not explicitly done on the global average center, we perform a calibration pass over the training set before evaluation, following Defazio et al. (2024)[1], which yields more reliable test performance. We report mean ± standard deviation from 3 repeat runs, and shaded regions in the plots indicate 95% confidence intervals. All experiments extend the decentralized training framework of Wang et al. (2025),[2], which overlaps decentralized communication with computation to hide communication latency. Additional training and evaluation details can be found in Appendix A.3.

This work focuses on DSGD with adaptive consensus, where the base optimizer is SGD without preconditioning. Appendix A.2.1 reports additional experiments with Adam (Kingma & Ba, 2014) as the base optimizer for Transformer models (Vaswani et al., 2017) on WMT-14 (Bojar et al., 2014), showing that adaptive consensus extends to preconditioned optimizers.

[1] `https://github.com/facebookresearch/schedule_free`
[2] `https://github.com/WangZesen/Decent-DP`

p	**Test Acc.** (%) ↑	**Train Loss** ↓	**Test Loss** ↓
0	96.04 ± 0.21	<u>0.0006</u> ± 0.0000	0.182 ± 0.007
1	96.36 ± 0.06	0.0007 ± 0.0000	0.160 ± 0.002
2	96.48 ± 0.13	0.0017 ± 0.0001	0.150 ± 0.002
3	<u>96.55</u> ± 0.10	0.0163 ± 0.0005	0.122 ± 0.002
4	96.41 ± 0.04	0.0347 ± 0.0010	<u>0.118</u> ± 0.001
5	96.19 ± 0.05	0.0475 ± 0.0013	0.121 ± 0.002

Table 1: Sensitivity analysis of parameter p in the WRN28-10 on CIFAR-10 experiments with 16 workers, one-peer ring topology, and $E_{\text{start}} = 100$. The best metric values are <u>underlined</u>.

E_{start}	**Test Acc.** (%) ↑	**Train Loss** ↓	**Test Loss** ↓
10	96.16 ± 0.06	0.0474 ± 0.0011	0.121 ± 0.003
50	96.31 ± 0.12	0.0395 ± 0.0006	0.121 ± 0.003
75	96.33 ± 0.09	0.0290 ± 0.0005	<u>0.119</u> ± 0.002
100	<u>96.55</u> ± 0.10	0.0163 ± 0.0005	0.122 ± 0.002
150	96.49 ± 0.08	0.0021 ± 0.0003	0.150 ± 0.002
175	96.21 ± 0.12	<u>0.0008</u> ± 0.0001	0.181 ± 0.003

Table 2: Sensitivity analysis of E_{start} in the WRN28-10 on CIFAR-10 experiments with 16 workers, one-peer ring topology, and $p = 3$. The best metric values are <u>underlined</u>.

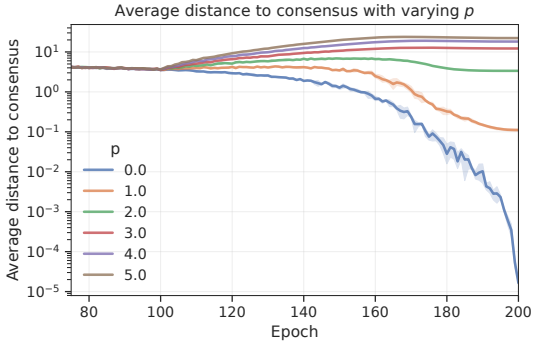


Figure 4: Average norm of consensus errors over epochs with varying p in the WRN28-10 on CIFAR-10 experiments with 16 workers, the one-peer ring topology, and $E_{\text{start}} = 100$.

4.1 Ablation study and sensitivity analysis

Start epoch E_{start} We introduce E_{start} as the epoch from which the adaptive consensus mechanism is activated. Before E_{start}, γ is set to 1 and DSGD-AC reduces to standard DSGD. In setups with more workers or weaker connectivity, decentralization already induces substantial consensus errors in early training, and reducing γ below 1 at this stage further weakens mixing, hindering loss optimization before the model has converged to a stable region. Activating too late, on the other hand, leaves too few epochs for the regularisation to act. Table 2 shows this two-sided sensitivity: for 16 workers on the one-peer ring topology, $E_{\text{start}} = 100$ gives the best test accuracy, with performance degrading at both extremes. As a practical heuristic, E_{start} should follow the warm-up phase and scale with worker count and topology: more workers and weaker connectivity require a later activation.

Parameter p Table 1 presents the sensitivity analysis with respect to p. The results indicate that the best test accuracy and test loss are achieved for $p \in [2, 4]$. Since DSGD-AC reduces to DSGD when $p = 0$, these experiments also serve as an ablation study of the adaptive consensus mechanism. With $p > 0$, DSGD-AC always achieves better performance than DSGD. Figure 4 presents the average norm of the consensus errors with various p. Consistent with the analysis in Section 3.4, the consensus errors persist for $p \geq 2$, and the regularization and generalization improvement become weak with a too-large p because of worse alignment.

Appendix A.2.3 includes the training curves of experiments in the sensitivity analysis.

4.2 Tuned results with various setups

In Table 3, we summarize the experiment results of DSGD-AC with tuned p and E_{start} based on Tables 1 and 2, and the comparisons with SGD and DSGD in various setups. The algorithms are applied on training WRN28-10 and WRN16-8 on the CIFAR-100 dataset. The number of workers varies from 8 to 32, and

Model	#W	Topo.	Algorithms (Test Accuracy (%) ↑ / Test Loss ↓ / Top-1 Eigenvalue ↓)		
			Synchronous SGD	Decentralized SGD	DSGD-AC (ours)
28-10	8	ring	80.00 ± 0.19 / 1.097 ± 0.014 / 4.53 ± 0.72	79.71 ± 0.13 / 1.108 ± 0.011 / 4.35 ± 1.19	<u>82.12</u> ± 0.12 / <u>0.897</u> ± 0.007 / 0.59 ± 0.02
		exp		79.98 ± 0.33 / 1.100 ± 0.018 / 4.60 ± 1.20	81.82 ± 0.08 / 0.911 ± 0.013 / 0.53 ± 0.04
		complete		79.62 ± 0.05 / 1.114 ± 0.014 / 4.13 ± 0.85	81.41 ± 0.31 / 0.953 ± 0.001 / <u>0.47</u> ± 0.09
	16	ring	79.79 ± 0.25 / 1.111 ± 0.021 / 6.75 ± 0.97	80.18 ± 0.14 / 1.050 ± 0.015 / 2.89 ± 1.01	<u>82.19</u> ± 0.17 / <u>0.846</u> ± 0.012 / 0.51 ± 0.02
		exp		80.18 ± 0.28 / 1.066 ± 0.003 / 3.31 ± 0.82	82.09 ± 0.03 / 0.886 ± 0.003 / 0.57 ± 0.16
		complete		79.81 ± 0.29 / 1.094 ± 0.007 / 4.11 ± 0.40	81.64 ± 0.10 / 0.934 ± 0.010 / <u>0.45</u> ± 0.03
	32	ring	79.90 ± 0.19 / 1.090 ± 0.019 / 4.02 ± 0.69	80.72 ± 0.34 / 0.950 ± 0.019 / 4.15 ± 0.61	<u>81.92</u> ± 0.13 / <u>0.831</u> ± 0.006 / 0.62 ± 0.12
		exp		80.56 ± 0.21 / 1.000 ± 0.006 / 3.58 ± 1.06	<u>81.92</u> ± 0.04 / 0.886 ± 0.011 / <u>0.37</u> ± 0.04
		complete		80.04 ± 0.12 / 1.060 ± 0.011 / 3.50 ± 1.09	81.58 ± 0.15 / 0.925 ± 0.001 / 0.42 ± 0.09
16-8	8	ring	79.19 ± 0.13 / 0.997 ± 0.007 / 6.81 ± 2.34	79.16 ± 0.36 / 0.986 ± 0.010 / 3.51 ± 0.60	<u>80.54</u> ± 0.31 / 0.910 ± 0.016 / 0.76 ± 0.03
		exp		78.90 ± 0.14 / 1.002 ± 0.012 / 4.71 ± 0.33	80.51 ± 0.08 / <u>0.902</u> ± 0.004 / <u>0.75</u> ± 0.04
		complete		79.12 ± 0.22 / 0.989 ± 0.012 / 8.89 ± 1.88	80.08 ± 0.20 / 0.931 ± 0.002 / 1.18 ± 0.76
	16	ring	79.02 ± 0.27 / 1.012 ± 0.014 / 6.32 ± 1.21	79.07 ± 0.22 / 0.979 ± 0.008 / 4.01 ± 1.36	<u>80.51</u> ± 0.11 / 0.861 ± 0.015 / 0.56 ± 0.01
		exp		79.02 ± 0.14 / 0.982 ± 0.011 / 4.91 ± 0.37	80.44 ± 0.08 / <u>0.889</u> ± 0.006 / 0.57 ± 0.02
		complete		79.03 ± 0.22 / 1.001 ± 0.012 / 5.32 ± 2.60	80.01 ± 0.06 / 0.919 ± 0.016 / <u>0.51</u> ± 0.01
	32	ring	78.90 ± 0.14 / 1.010 ± 0.009 / 8.41 ± 1.13	79.12 ± 0.24 / 0.976 ± 0.006 / 2.95 ± 0.39	80.17 ± 0.10 / <u>0.872</u> ± 0.005 / 0.72 ± 0.21
		exp		79.13 ± 0.04 / 0.974 ± 0.004 / 3.17 ± 0.19	<u>80.22</u> ± 0.16 / 0.921 ± 0.009 / <u>0.51</u> ± 0.03
		complete		79.04 ± 0.19 / 0.988 ± 0.014 / 2.81 ± 0.36	79.73 ± 0.13 / 0.955 ± 0.006 / 0.52 ± 0.01

Table 3: Performance comparison of synchronous SGD, decentralized SGD, and DSGD-AC. Dataset: CIFAR-100. Model: WRN28-10 and WRN16-8. Number of workers: 8, 16, 32. Topology: one-peer ring, one-peer exponential graph (Ying et al., 2021), and complete graph. $p = 3$ for all DSGD-AC experiments. E_{start} is set to 10, 100 and 150 for 8-worker, 16-worker and 32-worker setups, respectively. The best metric values for each distributed setup are <u>underlined</u>.

the topologies are one-peer ring, one-peer exponential, and complete graph. According to the sensitivity analysis, we choose $p = 3$ for all DSGD-AC experiments, and E_{start} =10, 100, and 150 for 8-worker, 16-worker, and 32-worker setups, respectively. Since the best sharpness metric remains an open question, we report the top-1 Hessian eigenvalue as a surrogate, which has been empirically shown to correlate strongly with generalization (Bisla et al., 2022; Luo et al., 2024).

Across all evaluated setups, DSGD-AC achieves consistently higher test accuracy and lower test loss than both synchronous SGD and decentralized SGD (even under their optimal setups). These results validate the effectiveness and robustness of DSGD-AC across models, topologies, and system scales. The corresponding training curves can be found in Appendix A.2.2. Finally, we validate that DSGD-AC specifically targets flatter minima as DSGD-AC significantly reduces the dominant Hessian eigenvalue compared to standard DSGD and synchronous SGD. This confirms that the adaptive consensus errors provide implicit sharpness-aware regularization similar to SAM (Foret et al., 2020), but without the 2× computational overhead required by explicit sharpness minimization methods (see Appendix A.2.4 for details).

4.3 Training efficiency

Table 4 shows the training time with WRN28-10 and CIFAR-100 using 1, 2, and 4 8×T4 nodes interconnected by 100Gbps InfiniBand. Thanks to the decentralized communication pattern, decentralized training can hide the communication time and provide more resilience to stragglers, which makes it more efficient than synchronous training even under the single-node setup (Lian et al., 2017; Assran et al., 2019; Wang et al., 2025). In these experiments, decentralized methods require only 0.78–0.85× the training time of synchronous SGD, and the relative speedup further improves as the number of workers increases. Compared with DSGD, DSGD-AC introduces negligible additional runtime overhead.

# Nodes	SGD	DSGD	DSGD-AC
1	91.50 ± 0.21 (×1.000)	77.69 ± 0.45 (×0.849)	78.11 ± 0.52 (×0.854)
2	48.38 ± 0.11 (×1.000)	39.09 ± 0.19 (×0.808)	39.69 ± 0.14 (×0.820)
4	25.78 ± 0.07 (×1.000)	20.08 ± 0.12 (×0.779)	20.28 ± 0.14 (×0.787)

Table 4: Training time (in minutes) of SGD, DSGD, and DSGD-AC in the WRN28-10 on CIFAR-100 experiments. The communication topology for decentralized methods is the one-peer ring topology. $\times x$ is the relative time compared with synchronous SGD in the same distributed setup.

5 Conclusion and discussion

Prior work on decentralized training has focused on reducing consensus errors, viewing them as an obstacle to convergence and generalization. This work argues the opposite: consensus errors, maintained at a controlled level, act as an implicit regularizer and improve generalization over both standard DSGD and synchronous SGD at negligible additional cost. The key theoretical contribution is a modal analysis of the consensus-error dynamics, which characterizes how the adaptive factor $\gamma^{(t)}$ balances radius and curvature profile through the stationary variance of graph-Hessian modes. This analysis introduces curvature exposure as the quantity that governs the regularization effect, and shows that DSGD-AC preserves it better than standard DSGD. Together with the scalability inherited from decentralized training, it positions DSGD-AC as a compelling alternative to synchronous SGD for distributed training.

Limitation and future work Two limitations merit attention. First, DSGD-AC achieves a weaker normalized Hessian alignment than standard DSGD, as shown in Figure 3. The analysis in Section 3.3 shows this does not undermine the curvature exposure argument, since the larger radius compensates. However, it suggests that the linear mixing operator with a decaying scalar factor is not optimal: a mixing operator designed to preserve alignment while maintaining radius could improve the regularization effect further. Second, the mechanism relies on the anisotropy of SGD gradient noise, which aligns naturally with the Hessian eigen space. Adaptive optimizers such as Adam (Kingma & Ba, 2014) modulate this noise, disrupting the anisotropy and weakening the alignment of consensus errors with sharp directions. Extending adaptive consensus to such optimizers is therefore non-trivial and remains an open problem. Whether DSGD-AC finds different basins of attraction than DSGD, and what role escape dynamics play in the observed generalization improvement, is a further open question.

Broader Impact Statement

The proposed method may contribute to more efficient distributed training by improving the generalization performance of decentralized optimization without requiring extra gradient evaluations or substantial memory overhead. Such efficiency gains could reduce communication costs and, in some settings, lower the computational resources needed to reach a target accuracy. However, efficiency improvements may also encourage larger-scale training runs, potentially offsetting environmental benefits. We therefore view DSGD-AC as a tool for improving the quality-efficiency trade-off, rather than as a standalone solution to the environmental costs of modern deep learning.

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.

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.

Gerard Ben Arous, Reza Gheissari, Jiaoyang Huang, and Aukosh Jagannath. High-dimensional sgd aligns with emerging outlier eigenspaces. In International Conference on Learning Representations, volume 2024, pp. 47732–47778, 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.

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.

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.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

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.

Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. Journal of research of the National Bureau of Standards, 45(4):255–282, 1950.

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.

Sidak Pal Singh, Gregor Bachmann, and Thomas Hofmann. Analytic insights into structure and rank of neural network hessian maps. Advances in Neural Information Processing Systems, 34:23914–23927, 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.

Lei Wu, Mingze Wang, and Weijie J. Su. The alignment property of SGD noise and how it helps select flat minima: A stability analysis. In Advances in Neural Information Processing Systems, volume 35, pp. 16843–16857, 2022. URL https://arxiv.org/abs/2207.02628.

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.

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.

Liu Ziyin, Kangqiao Liu, Takashi Mori, and Masahito Ueda. Strength of minibatch noise in SGD. In International Conference on Learning Representations, 2022. URL `https://arxiv.org/abs/2102.05375`.

A Appendix

A.1 Proofs and derivations

A.1.1 Proof of Lemma 1

Proof. Since $P\mathbf{1} = 0$ and $P\Delta^{(t-1)} = \Delta^{(t-1)}$ (as $\sum_i \delta_i^{(t-1)} = 0$ by the definition of Δ), applying Assumption 4 to the disagreement recursion Equation 11 and invoking Assumption 5 ($H^{(t)} = H_f$, $R^{(t)} = 0$) gives

$$\begin{aligned}\Delta^{(t)} &= (I - \gamma^{(t)}L)\Delta^{(t-1)} - \alpha^{(t)}PG^{(t)} \\ &= (I - \gamma^{(t)}L)\Delta^{(t-1)} - \alpha^{(t)}P(\mathbf{1}\nabla f(\bar{x}^{(t-1)})^\top + \Delta^{(t-1)}H^{(t)} + R^{(t)} + \Xi^{(t)}) \\ &= (I - \gamma^{(t)}L)\Delta^{(t-1)} - \alpha^{(t)}P(\Delta^{(t-1)}H^{(t)} + R^{(t)} + \Xi^{(t)}) \\ &= (I - \gamma^{(t)}L)\Delta^{(t-1)} - \alpha^{(t)}\Delta^{(t-1)}H_f - \alpha^{(t)}P\Xi^{(t)}.\end{aligned} \tag{22}$$

Multiplying Eq. (22) by u_k on the right and using $H_f u_k = \lambda_k u_k$ gives

$$\Delta^{(t)}u_k = (I - \gamma^{(t)}L - \alpha^{(t)}\lambda_k I)\Delta^{(t-1)}u_k - \alpha^{(t)}P\Xi^{(t)}u_k. \tag{23}$$

For every disagreement mode $j \in \{1, \dots, n-1\}$, $Pv_j = v_j$ and $Lv_j = \mu_j v_j$. Multiplying by $v_j^\top$ on the left yields and setting $z_{j,k}^{(t)} := v_j^\top \Delta^{(t)} u_k$, $\xi_{j,k}^{(t)} := v_j^\top \Xi^{(t)} u_k$ yields the claimed recursion. □

A.1.2 Proof of Theorem 1

Proof. For fixed α and γ, Lemma 1 gives

$$z_{j,k}^{(t)} = a_{j,k} z_{j,k}^{(t-1)} - \alpha \xi_{j,k}^{(t)}, \quad a_{j,k} := 1 - d_{j,k} \tag{24}$$

Since $z_{j,k}^{(t-1)}$ is $\mathcal{F}_{t-1}$-measurable and $\mathbb{E}[\xi_{j,k}^{(t)} \mid \mathcal{F}_{t-1}] = 0$, squaring and taking conditional expectation yields

$$\mathbb{E}[(z_{j,k}^{(t)})^2 \mid \mathcal{F}_{t-1}] = a_{j,k}^2 (z_{j,k}^{(t-1)})^2 + \alpha^2 \sigma_{j,k}^2. \tag{25}$$

The condition $0 < d_{j,k} < 2$ is equivalent to $|a_{j,k}| < 1$, so the scalar autoregressive process is stable and admits a unique stationary distribution with finite second moment. In particular, imposing stationarity $\mathbb{E}[(z_{j,k}^{(t)})^2] = \mathbb{E}[(z_{j,k}^{(t-1)})^2] = V$ implies that

$$V = a_{j,k}^2 V + \alpha^2 \sigma_{j,k}^2. \tag{26}$$

Since $|a_{j,k}| < 1$ implies $1 - a_{j,k}^2 \neq 0$, this equation has the unique solution

$$V = \frac{\alpha^2 \sigma_{j,k}^2}{1 - a_{j,k}^2} = \frac{\alpha^2 \sigma_{j,k}^2}{d_{j,k}(2 - d_{j,k})}, \tag{27}$$

where the last equality uses $1 - a_{j,k}^2 = (1 - a_{j,k})(1 + a_{j,k}) = d_{j,k}(2 - d_{j,k})$. □

A.1.3 Derivation for Remark 1

Derivation. Since $0 < d_{j,k} \leq d_{\max} < 2$, the bounds $2 - d_{\max} \leq 2 - d_{j,k} \leq 2$ applied to Theorem 1 give

$$\frac{\alpha^2 \sigma_{j,k}^2}{2 d_{j,k}} \leq \mathbb{E}[z_{j,k}^2] \leq \frac{\alpha^2 \sigma_{j,k}^2}{(2 - d_{\max}) d_{j,k}}. \tag{28}$$

For mixing-dominated modes, $|\alpha \lambda_k| \leq \rho \gamma \mu_j$ implies

$$(1 - \rho)\gamma \mu_j \leq d_{j,k} \leq (1 + \rho)\gamma \mu_j. \tag{29}$$

Together with $d_{j,k} \leq d_{\max} < 2$, this makes $d_{j,k}(2 - d_{j,k}) = \Theta(\gamma \mu_j)$. Substituting into Eq. (13) proves the low-curvature claim. For $\lambda_k > 0$, the denominator is

$$d_{j,k}(2 - d_{j,k}) = \Theta(d_{j,k}) = \Theta(\gamma \mu_j + \alpha \lambda_k), \tag{30}$$

where the constants depend on $d_{\max}$. The same identity remains true for negative λ_k whenever $0 < d_{j,k} < 2$, but negative curvature decreases $d_{j,k}$ relative to $\gamma \mu_j$. □

A.1.4 Proof of Corollary 1

Proof. By Parseval's identity for the orthonormal bases V and U,

$$\|\Delta\|_F^2 = \sum_{j=1}^{n-1} \sum_{k=1}^{d} z_{j,k}^2. \tag{31}$$

Remark 1 gives

$$\mathbb{E}[z_{j,k}^2] = \Theta\left(\frac{\alpha^2 \sigma_{j,k}^2}{\gamma \mu_j}\right) \tag{32}$$

for every $(j,k) \in \mathcal{S}_{\text{low}}$. Summing over the active low-curvature modes yields

$$\sum_{(j,k) \in \mathcal{S}_{\text{low}}} \mathbb{E}[z_{j,k}^2] = \Theta\left(\frac{\alpha^2}{\gamma} \sum_{(j,k) \in \mathcal{S}_{\text{low}}} \frac{\sigma_{j,k}^2}{\mu_j}\right) = \Theta\left(\frac{\alpha^2}{\gamma} M_{\text{low}}(\alpha, \gamma)\right). \tag{33}$$

If $M_{\text{low}}(\alpha, \gamma)$ is bounded above and below by positive constants in the local window, this reduces to $\Theta(\alpha^2/\gamma)$. For every $(j,k) \in \mathcal{S}_{\text{high}}^+$, $\lambda_k \geq c > 0$, so

$$d_{j,k} = \gamma \mu_j + \alpha \lambda_k \geq \alpha c. \tag{34}$$

Since $d_{j,k} \leq d_{\max} < 2$, Eq. (13) gives

$$\mathbb{E}[z_{j,k}^2] \leq C \frac{\alpha^2 \sigma_{j,k}^2}{\alpha c} = O(\alpha) \tag{35}$$

for a constant C independent of α. Summing over the finite set $\mathcal{S}_{\text{high}}^+$ preserves the $O(\alpha)$ rate. If $M_{\text{low}}(\alpha, \gamma)$ remains non-negligible and $\alpha^2/\gamma = \Omega(\alpha)$, the low-curvature contribution decays no faster than this high-curvature upper bound. □

A.1.5 Proof of Corollary 2

Proof. If $\gamma = \alpha^p$ up to a constant factor, the mixing-dominated condition

$$|\alpha\lambda_k| \leq \rho\gamma\mu_j \tag{36}$$

is equivalent to $|\lambda_k| \leq C\rho\mu_j\alpha^{p-1}$ for a fixed constant $C > 0$. For such modes, Remark 1 gives

$$\mathbb{E}[z_{j,k}^2] = \Theta\left(\frac{\alpha^2\sigma_{j,k}^2}{\alpha^p\mu_j}\right) = \Theta\left(\alpha^{2-p}\sigma_{j,k}^2/\mu_j\right). \tag{37}$$

For $p = 2$ this is constant order in α for each active mixing-dominated mode; for $p > 2$ it grows in the linear stationary approximation for each such mode. The condition $|\lambda_k| \leq O(\alpha^{p-1})$ shrinks as α decreases, so the statement applies only to modes that remain active in this mixing-dominated regime. □

A.2 Additional experiments and details

A.2.1 Decentralized Adam with adaptive consensus

We also validate the idea of controlling consensus errors on transformer models by simply replacing the local update with the Adam optimizer (Kingma & Ba, 2014). DSGD-AC is then adapted to DAdam-AC as in Algorithm 2.

Algorithm 2: Decentralized Adam with adaptive consensus (DAdam-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**

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

$m_i^{(t)} = \beta_1 m_i^{(t)} + g_i^{(t)}$

$v_i^{(t)} = \beta_2 v_i^{(t)} + g_i^{(t)} \odot g_i^{(t)}$

$\hat{m}_i^{(t)} = m_i^{(t)}/(1-\beta_1^t)$

$\hat{v}_i^{(t)} = v_i^{(t)}/(1-\beta_1^t)$

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

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

end

We train Transformer (the big variant, ∼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 5.

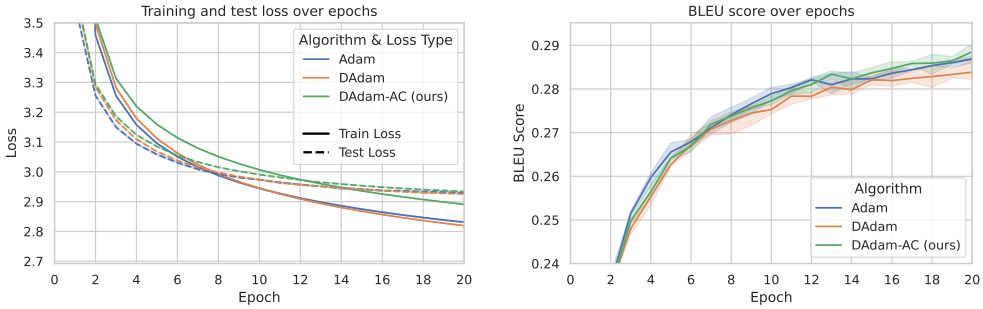


Figure 5: Transformer (big) on WMT14 English-to-German. **Left**: Losses on training set. **Right**: BLEU scores on the test set.

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	28.89 ± 0.17	2.9205 ± 0.0020	2.8456 ± 0.0016

Table 5: Performance comparison of DAdam, Adam, and DAdam-AC on neural machine translation with the transformer model.

The results show that DAdam-AC can outperform other baselines on the translation quality metric and the test loss. The adaptive consensus enhances the decentralized Adam, but the improvement is relatively marginal compared with the image classification task. We believe further improvement is possible by taking adaptive consensus into account when designing the optimizer or the mixing operator (see the discussion in Section 5).

A.2.2 Training curves

Figures 6 to 11 present the training curves corresponding to the experiments in Table 3. The figures show the impact of the adaptive consensus after the epoch it is activated, and the trend that DSGD-AC trades training loss for better test loss and test accuracy. Moreover, from the figures, one can observe that there is an increasing trend in test loss in the last 30-50 epochs of training. Based on the analysis in Section 3.4, a heuristic explanation is that the regularization becomes weaker on the top eigenvalues due to the worse alignment at the late-training phase, which hinders the test performance.

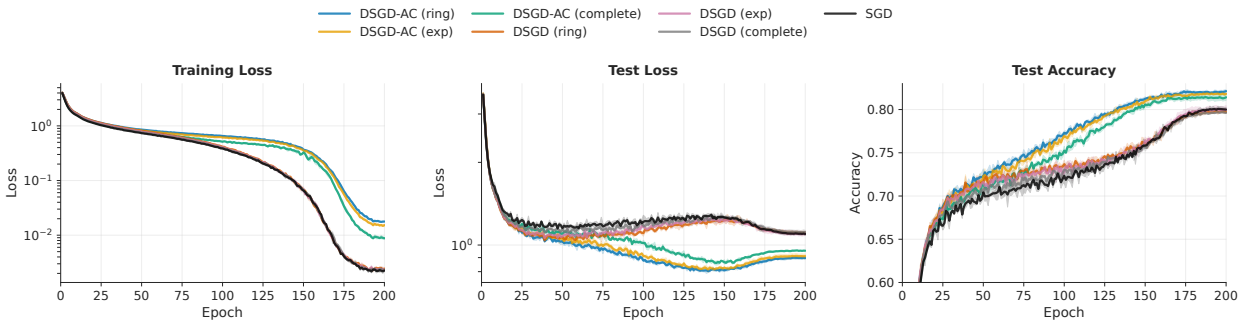


Figure 6: WRN28-10, CIFAR100, 8 workers

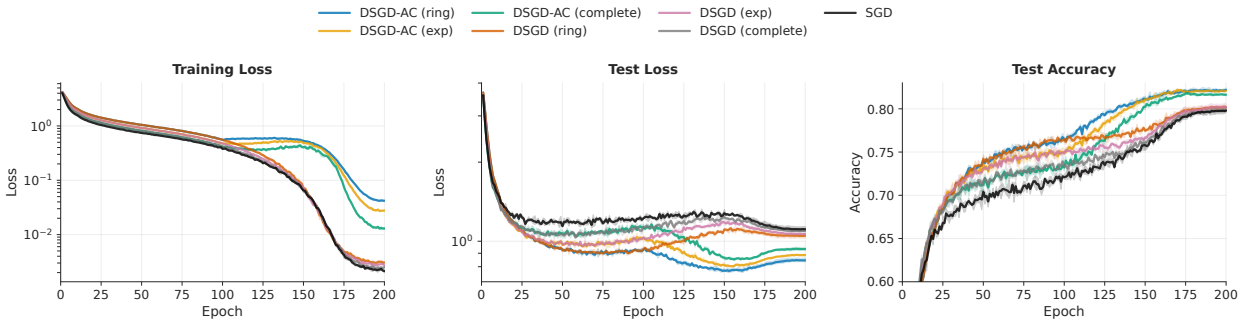


Figure 7: WRN28-10, CIFAR100, 16 workers

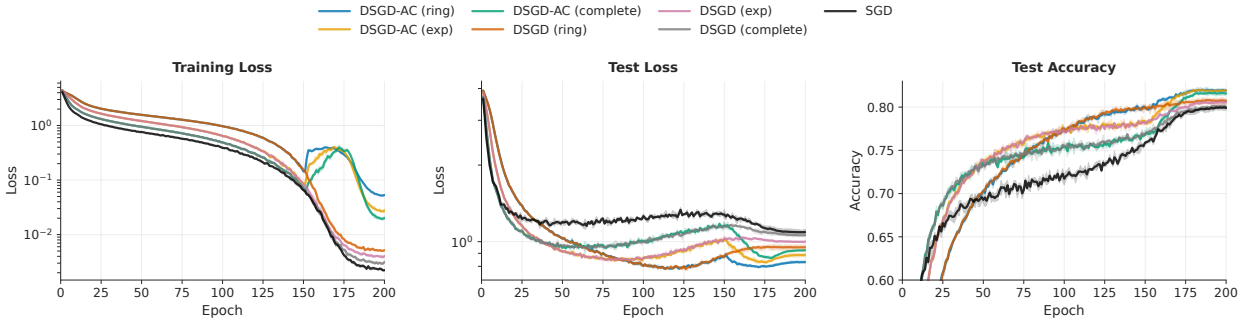


Figure 8: WRN28-10, CIFAR100, 32 workers

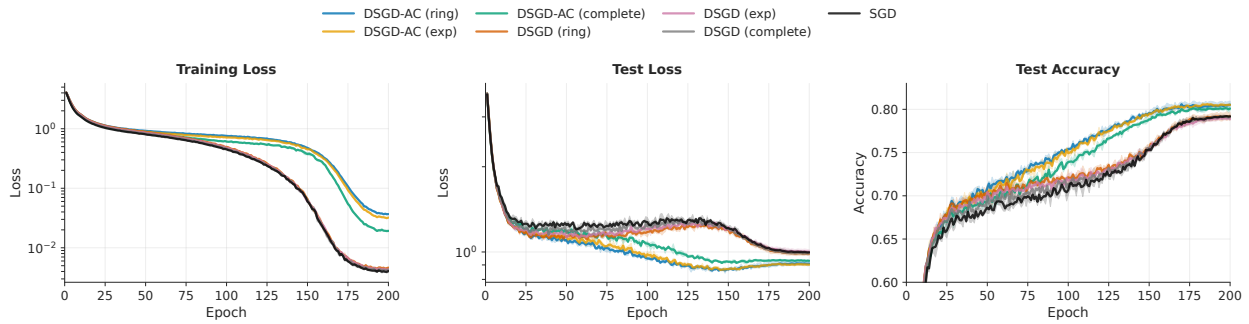


Figure 9: WRN16-8, CIFAR100, 8 workers

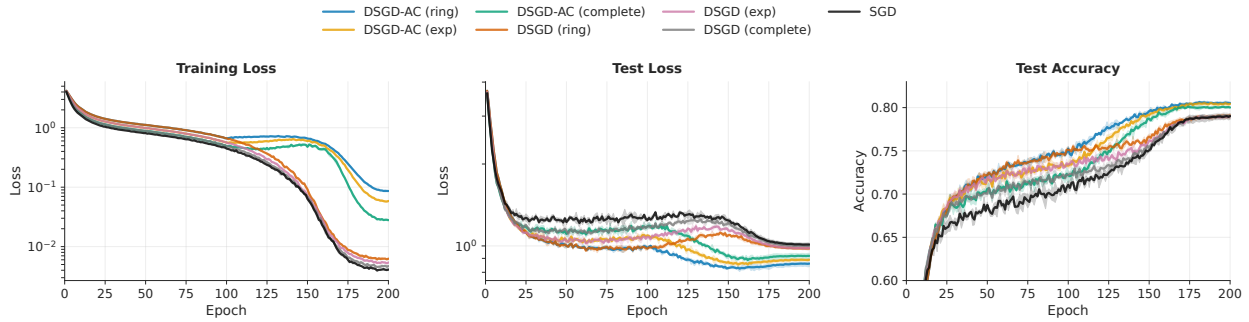


Figure 10: WRN16-8, CIFAR100, 16 workers

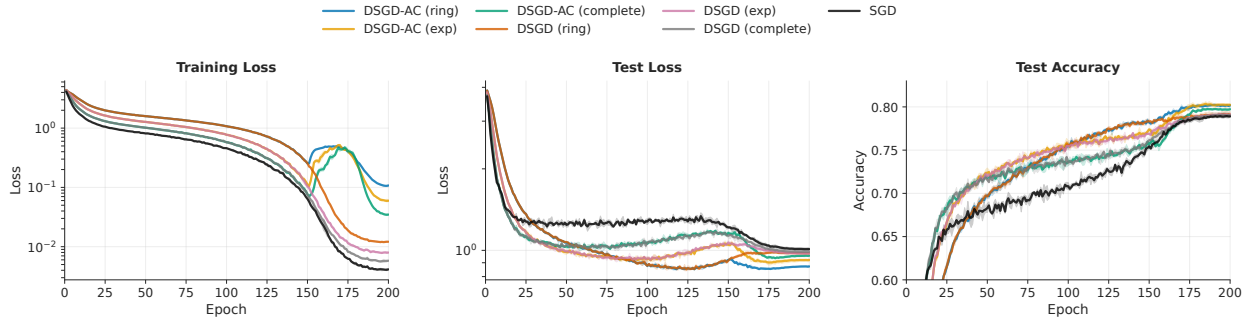


Figure 11: WRN16-8, CIFAR100, 32 workers

A.2.3 Sensitivity analysis

Figures 12 and 13 show the curves of the test accuracy and the test loss of the sensitivity analysis experiments.

For p, a greater-than-zero p gives better test performance, while the benefits diminish as $p > 3$.

E_{start} is the epoch from which the adaptive consensus is activated. We set $\gamma = 1$ before E_{start}, and make $\alpha_{\max}$ equal to the maximal learning rate among the iterations where DSGD-AC is activated.

The results show that $E_{\text{start}} = 100$ gives the best performance on test accuracy, and, for $E_{\text{start}} \in \{10, 50, 75, 100\}$, they give similar test loss performance, and a too-late activation of the adaptive consensus ($E_{\text{start}} = \{150, 175\}$) leaves too few iterations for DSGD-AC to take effect.

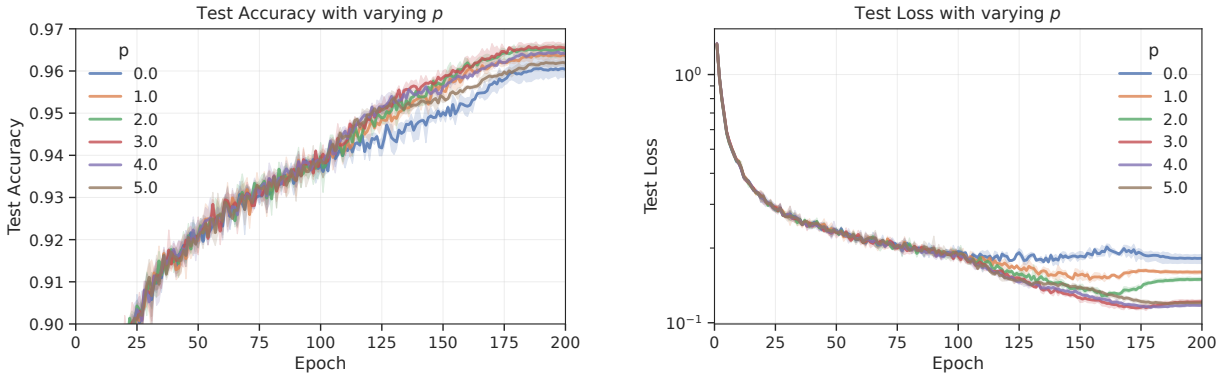


Figure 12: DSGD-AC with WRN28-10 on CIFAR10 with varying p and $E_{\text{start}} = 100$.

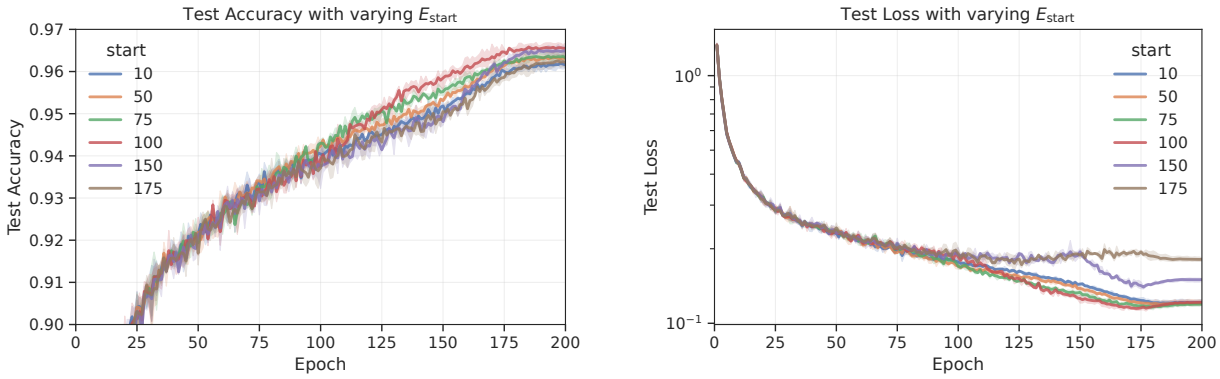


Figure 13: DSGD-AC with WRN28-10 on CIFAR10 with varying E_{start} and $p = 3$.

A.2.4 Sharpness-aware minimization

We implement SAM (Foret et al., 2020) and report its results with 8 and 16 workers (1 and 2 8×T4 nodes) for reference. We follow their work to use $\rho = 0.05$ in all the experiments. The results show that SAM achieves better metric values than DSGD-AC. However, it should be noted that SAM introduces 2× computation cost, and it is not numerically stable under the mixed precision training in our setup (the loss becomes NaN in around 30 epochs with AMP in fp16). Therefore, it takes $\sim 4\times$ training time as the synchronous SGD in the 8-worker and 16-worker setups under the same number of iterations. Even under a roughly the same FLOP budget (100 epochs), SAM still takes around 2× training time, and the accuracies are comparable to DSGD-AC.

# Nodes	Epochs	Test Acc. (%) ↑	Test Loss ↓	Train Loss ↓	Training Time (min) ↓
1	200	83.04 ± 0.22	0.700 ± 0.007	0.0519 ± 0.0003	382.14 ± 8.32
2	200	83.00 ± 0.03	0.697 ± 0.004	0.0517 ± 0.0003	193.97 ± 1.18
1	100	82.49 ± 0.09	0.692 ± 0.002	0.0819 ± 0.0007	196.10 ± 1.33
2	100	82.48 ± 0.10	0.685 ± 0.005	0.0836 ± 0.0007	96.61 ± 0.78

Table 6: Performance of SAM on image classification tasks with WRN28-10 on CIFAR-100.

A.3 Hyperparameter details

A.3.1 Hyperparameters for image classification experiments with WRN on CIFAR10/100

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

Category	Setting
General	
Number of epochs	200
Global batch size	128 for 8-worker setup and it is linearly scaled with the number of workers.
Learning rate scheduler	Linearly warm-up to $\alpha_{\max}$ in the first 10 epochs, followed by the cosine annealing until the end. $\alpha_{\max} = 0.1$ for 128 batch size, and it is linearly scaled with the batch size.
Base optimizer	SGD with momentum $\beta = 0.9$ and weight decay 5×10^{-4}.
Data shuffle	Randomly shuffled and split into N local datasets each epoch.
Decentralized training	
Number of workers	8, 16, and 32
Communication topology	One-peer ring (alternating between neighbors $i-1$ and $i+1$ across iterations), one-peer exponential Ying et al. (2021), and complete graph (global all-reduce among all workers).
DSGD-AC parameters	Exponent $p = 3$; $\gamma = 1$ before E_{start}.
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. Note that we also apply the calibration to synchronous SGD for a fair comparison.

A.3.2 Hyperparameters for neural machine translation experiments with Transformer on WMT14

The selection of hyperparameters follows the original paper (Vaswani et al., 2017), and our baseline implementation matches its performance. We tuned the learning rate and (β_1, β_2) separately for decentralized methods.

Category	Setting
General	
Number of epochs	20
Global batch size	∼50k tokens including both source and target texts
Learning rate scheduler	Linear warm-up to 5×10^{-4} over the first 4000 iterations, then decay as $\alpha_0 \cdot (4000/t)^{0.5}$ (t is the iteration index). $\alpha_0 = 0.0005$ for centralized Adam, and $\alpha_0 = 0.0013$ for decentralized methods.
Base optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98$) for centralized Adam, and ($\beta_1 = 0.974$, $\beta_2 = 0.999$) for decentralized methods.
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 $p = 2$ (tuned based on experiments); $\gamma = 1$ during warm-up.
Normalization	Since only layer normalization is used, no calibration is needed.

A.3.3 Evaluation details

Eigenvalue evaluation We use the Lanczos algorithm (Lanczos, 1950) for efficient evaluation of the max and min eigenvalues of the Hessian. To be specific, the loss function is

$$F(x) = \frac{1}{B}\sum_{b=1}^{B} f(x; s_b) \tag{38}$$

where s_b denotes a batch of samples from the training set, $\mathcal{S}$ denotes the total number of samples in the training set, and $\{s_1, \cdots, s_B\}$ collectively cover the entire training set. To exclude the impact of mismatched statistics in the batch normalization layers and to reflect the actual training loss, we evaluate the Hessian-vector product in training mode and control the random seed to ensure that the sample order, batching, and data augmentation are identical across all evaluations. In our evaluation, we fix the mini-batch size to 32.

We fix the number of Lanczos iterations to 30.

Alignment evaluation The Hessian quadratic with any vector can be computed via the dot product between the consensus error and the Hessian vector product, and the Hessian vector product (HVP) is computed using the Pearlmutter trick without materializing the Hessian, which is supported out-of-the-box by the automatic differentiation in PyTorch.

$$D := \nabla F(x)^\top v, \quad \frac{\partial D}{\partial x} = H(x)v \tag{39}$$

where $F(x)$ is defined in the same way as in Eq. (38), and, since $F(x)$ is a linear combination of loss functions on batches, the HVP can be computed in a distributed way for better efficiency.

We use 50 Hutchinson trace probes to evaluate the alignment between the Hessian and random directions in our showcase runs, and the standard deviations are on the order of 10^{-7}.

Gradient noise alignment evaluation The gradient noises and their alignment with the Hessian are computed by

$$\xi_i := \nabla f(x; s_i) - \frac{1}{B}\sum_{b=1}^{B} \nabla f(x; s_b), \quad A_{\text{grad}} := \frac{1}{B}\sum_{b=1}^{B} \xi_b^\top H \xi_b / \lambda_1(H) \tag{40}$$

where the first step is simply computing the difference between the mini-batch loss and the full-batch loss, and the second step is computed via HVP as in the alignment evaluation.