DIMAT: Decentralized Iterative Merging-And-Training for Deep Learning Models

Nastaran Saadati[1] nsaadati@iastate.edu
Minh Pham[2] mp5847@nyu.edu
Nasla Saleem[1] nasla@iastate.edu
Joshua R. Waite[1] jrwaite@iastate.edu
Aditya Balu[1] baditya@iastate.edu
Zhanong Jiang[1] zhjiang@iastate.edu
Chinmay Hegde[2] chinmay.h@nyu.edu
Soumik Sarkar[1] soumiks@iastate.edu

[1]Iowa State University, Ames; [2]New York University, New York

Abstract

Recent advances in decentralized deep learning algorithms have demonstrated cutting-edge performance on various tasks with large pre-trained models. However, a pivotal prerequisite for achieving this level of competitiveness is the significant communication and computation overheads when updating these models, which prohibits the applications of them to real-world scenarios. To address this issue, drawing inspiration from advanced model merging techniques without requiring additional training, we introduce the Decentralized Iterative Merging-And-Training (DIMAT) paradigm—a novel decentralized deep learning framework. Within DIMAT, each agent is trained on their local data and periodically merged with their neighboring agents using advanced model merging techniques like activation matching until convergence is achieved. DIMAT provably converges with the best available rate for non-convex functions with various first-order methods, while yielding tighter error bounds compared to the popular existing approaches. We conduct a comprehensive empirical analysis to validate DIMAT's superiority over baselines across diverse computer vision tasks sourced from multiple datasets. Empirical results validate our theoretical claims by showing that DIMAT attains faster and higher initial gain in accuracy with independent and identically distributed (IID) and non-IID data, incurring lower communication overhead. This DIMAT paradigm presents a new opportunity for the future decentralized learning, enhancing its adaptability to real-world with sparse and light-weight communication and computation.

1. Introduction

Faced with various large-scale deep learning tasks [5, 19, 40], researchers and practitioners have made considerable efforts to advance numerous decentralized training algorithms [6]. These also benefit from the rapid development of hardware technologies [13, 41]. Decentralized deep learning algorithms have demonstrated compellingly cutting-edge performance, nearly matching their centralized counterparts [17, 26, 47, 59]. The most popular decentralized learning algorithms are first-order methods such as decentralized stochastic gradient descent (SGD) [10, 25, 46],

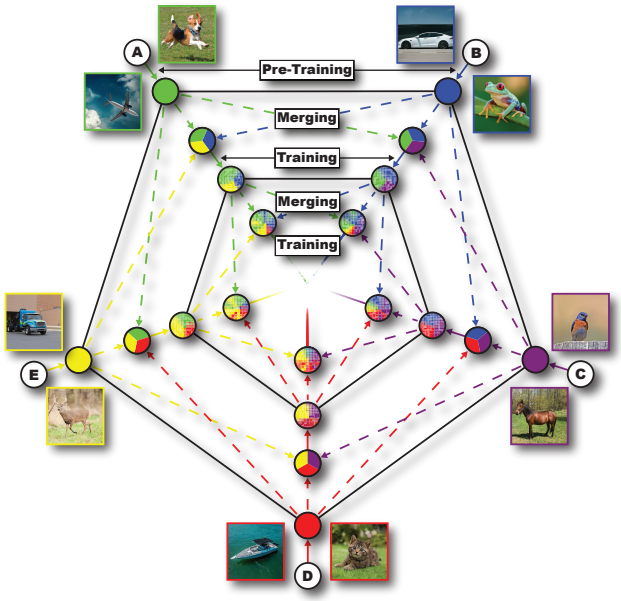


Figure 1. Illustration of DIMAT with ring topology for 5 agents. Agents, denoted by capital letters, undergo pre-training on two unique classes from CIFAR-10 each (solid lines). Subsequently, adjacent agents merge, forming tri-colored nodes (dashed lines). Updated agents then train on the original datasets (solid lines) with a slight increase in dataset colors. This merging and training cycle repeats for specified iterations until final fine-tuning.

Method	Convergence Rate
CD(M)SGD [58]	$\mathcal{O}(\sqrt{\frac{1}{NK}} + \frac{N}{(1-\rho)K} + \frac{N}{(1-\sqrt{\rho})^2 K})$
SGP [3]	$\mathcal{O}(\sqrt{\frac{1}{NK}} + \frac{N}{(1-\rho)^2 K} + \frac{N}{(1-\rho)^2 K^{1.5}})$
CGA [17]	$\mathcal{O}(\sqrt{\frac{1}{NK}} + \frac{N}{(1-\rho)K} + \frac{\sqrt{N}}{K^{1.5}} + \frac{N}{(1-\sqrt{\rho})^2 K})$
DIMAT	$\mathcal{O}(\sqrt{\frac{1}{NK}} + \frac{N}{(1-\rho')K} + \frac{N}{(1-\sqrt{\rho'})^2 K})$

Table 1. Comparison between different methods. N: # of agents, K: the number of iterations, $0 < \rho, \rho' < 1$: positive constant related to doubly stochastic matrices and $\rho' \leq \rho$.

their momentum accelerated variants [4, 57, 59], and more recently developed decentralized adaptive gradient methods [9, 36], which all provably show convergence with a sublinear rate and empirically exhibit appealing performance.

A prerequisite for delivering competitive results is the significant communication and computation overheads, particularly when the models are large such as VGG [14] and ResNet [49]. However, such an requirement extremely prohibits the use of decentralized learning in real-world scenarios. Another concern is the slower performance gain when they are utilized with sparser topology networks, which results in more pronounced communication and computation overheads [8]. Some recent works [11, 30, 55, 62] have attempted to address the above issues; however, they still remain critically challenging when local agents have widely distinct tasks.

Inspired by a recent line of works in deep learning — model merging [1, 12, 38, 45, 48] — our paper proposes a novel decentralized deep learning paradigm, that we call Decentralized Iterative Merging-And-Training (DIMAT). Specifically, within DIMAT, each agent is first trained on their local data and then periodically *merged* with their neighboring agents using advanced model merging techniques. Such a merging-and-training manner is iterated until the convergence is reached. Our method is different from other decentralized deep learning methods significantly driven by vanilla weight averaging [10, 25, 32]; instead, it leverages modulo permutation symmetries from model merging to allow local agents reaching a better "consensus regime" by enlarging the spectral gap; this eventually leads to a smaller optimality gap. Please see Figure 1 for the schematic illustration of DIMAT.

In addition, DIMAT is found to speed up performance gain at the early phase of optimization with lower communication cost. Our contributions are as follows:

- We propose and develop DIMAT, a novel decentralized deep learning framework with periodical model merging in the communication protocol. For the local training strategy, DIMAT can be equipped with different first-order methods flexibly. The model merging frequency can even be adjusted in diverse scenarios to reduce the communication overheads.
- We theoretically show that DIMAT provably converges with a sublinear rate to a stationary point for nonconvex functions, while yielding a tighter error bound and maintaining linear speed up, compared to the popular existing approaches. The theory also implies the faster initial performance gain due to the larger spectral gap compared to existing algorithms.
- A comprehensive empirical analysis is conducted to validate DIMAT's efficiency and superiority over baselines across IID and non-IID data distributions sourced from three benchmark datasets, including CIFAR-10, CIFAR-100, and Tiny ImageNet datasets by using two popular deep neural network models. See Table 1 for the comparison between the proposed and existing algorithms.

2. Related Work

Decentralized Learning: Several decentralized learning algorithms have demonstrated performance comparable to centralized counterparts on standard vision datasets. In their comprehensive investigation, Lian et al. [32] conducted a case study on decentralized parallel stochastic gradient descent (D-PSGD), combining stochastic gradient descent with a gossip averaging algorithm [56]. Jiang et al. introduced consensus-based distributed SGD (CDSGD) exploring collaborative deep learning in fixed topology networks, contributing insights into decentralized cooperation [25]. The extension of D-PSGD to directed and time-varying graphs introduced Stochastic Gradient Push (SGP) [3], while a momentum version, Decentralized Momentum Stochastic Gradient Descent (DMSGD), was proposed in [4]. Fotouhi et al. proposed an algorithm called minimum connected Dominating Set Model Aggregation (DSMA) to address communication overhead issues [18]. However, a critical assumption for achieving state-of-the-art performance in these decentralized algorithms is that the data is IID across agents [3, 18, 25]. Although, Khawatmi et al. delved into decentralized clustering and linking, shedding light on challenges posed by heterogeneous settings

and node assignment within clusters [29]. Recent efforts aim to bridge the performance gap between IID and non-IID data in decentralized setups [2, 17, 39, 50, 51]. Lin et al. proposed a Quasi-Global momentum-based decentralized deep learning approach for heterogeneous data [33]. Tang et al. introduced the D^2 algorithm, an extension of D-PSGD tailored for non-IID data, while Nadiradze et al. proposed SwarmSGD, utilizing random interactions between agents in a graph for consensus [35, 50]. Esfandiari et al. introduced Cross-Gradient Aggregation (CGA) and its compressed variant (CompCGA) for decentralized learning on entirely non-IID data, asserting superior performance [17]. However, these techniques incur higher communication costs compared to standard decentralized algorithms such as DSGD. Federated learning methods, such as the one proposed by McMahan et al. [34], offer a practical solution for training deep networks by iteratively averaging models, thus reducing communication costs compared to synchronous stochastic gradient descent. Federated Multi-Task Learning [43], demonstrated the natural suitability of multi-task learning to handle the statistical challenges of federated learning settings. Our primary aim is to leverage multi-task learning and decentralized learning on IID and non-IID data with minimal computational and communication overhead.

Model Merging: Recently, model merging has emerged as a technique that integrates the parameters or predictions of multiple deep-learning models into a single one, requiring minimal computation. This process exploits the proximity of models with the same pre-trained weights in the same error basin [37]. Numerous studies [21, 24, 52, 54] have utilized this characteristic to average the weights of a model during training. The merging process can enhance single-task performance for models designed for the same task [53] or create a multi-task solution for models targeting different tasks [15, 22, 54]. Various merging methods such as linear interpolation, and task arithmetic [22, 27] have also been proposed. However, for models without shared pre-trained weights, weight permutation may be necessary before merging [1, 16, 28]. In our approach, we leverage modulo permutation symmetries from model merging, enabling local agents to reach a more robust "consensus regime". This integrated approach aims to address challenges arising from heterogeneous data distributions, providing an effective solution with minimal computational and communication overhead.

3. Methodology

In this section, we first present some preliminaries regarding one model merging technique called activation matching, and then formulate the generic optimization problem in the decentralized learning setting, followed by the proposed algorithms.

3.1. Preliminaries: Activation Matching

In this section, we present the activation matching methodologies, extending the principles established by Ainsworth et al. [1]. These methodologies form the cornerstone of our decentralized model merging approach, inspired by the concept that models acquiring similar features are likely to perform similar tasks [31]. Consider two multi-layer perceptron models, M_1 and M_2. Given activations for each model, we aim to associate each unit in M_1 with a unit in M_2. This is accomplished by fitting the linear relationship into the regression framework, constraining a matrix of cross-correlation coefficients to solutions in the set of all symmetric permutation matrices, $S_d \in \mathbb{R}^{d\times d}$. For the ℓ-th layer, let $\mathbf{Z}^{(M_1)} \in \mathbb{R}^{d\times s}$ and $\mathbf{Z}^{(M_2)} \in \mathbb{R}^{d\times s}$ denote the cross-correlation of the activations of models M_1 and M_2, respectively, where s signifies the number of all training data points in models. We aim to find a permutation matrix $\mathbf{P}_\ell \in \mathbb{R}^{d\times d}$ that minimizes the Frobenius norm of the difference between $\mathbf{Z}^{(M_1)}$ and the permuted $\mathbf{Z}^{(M_2)}$:

$$\begin{aligned}\mathbf{P}_\ell &= \arg\min_{\mathbf{P}\in S_d} \sum_{p=1}^{s} \|\mathbf{Z}^{(M_1)}_{:,p} - \mathbf{P}\mathbf{Z}^{(M_2)}_{:,p}\|^2 \\ &= \arg\max_{\mathbf{P}\in S_d} \langle \mathbf{P}, \mathbf{Z}^{(M_1)}(\mathbf{Z}^{(M_2)})^\top \rangle_F.\end{aligned} \tag{1}$$

Please note that the dimension of $\mathbf{P}$ will be expanded in the decentralized learning setting in the rest of the paper. This problem constitutes a "linear assignment problem" (LAP), for which efficient algorithms exist. Once LAP is solved for each layer, we proceed to permute the parameters of model M_2 to align them with those of model M_1:

$$\mathcal{W}'_\ell = \mathbf{P}_\ell \mathcal{W}^{M_2}_\ell \mathbf{P}^\top_{\ell-1}, \quad b'_\ell = \mathbf{P}_\ell b^{M_2}_\ell,$$

where $\mathcal{W}'_\ell$ and $\mathcal{W}^{M_2}_\ell$are weight matrices, b'_ℓ and $b^{M_2}_\ell$ are bias vectors. We will utilize the activation matching method in the communication phase of our DIMAT framework for decentralized model merging. This approach represents a more sophisticated strategy compared to the basic averaging technique employed in prior methods, such as consensus-based decentralized SGD (CDSGD) or its momentum variant [25, 58]. Throughout the paper, we still use model merging instead of activation matching to better fit the DIMAT name. More detailed information about this method has been provided in the supplementary material.

3.2. Problem Formulation

Consider a network involving N agents and denote by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ the connected topology, where $\mathcal{V} = \{1, 2, ..., N\}$ and $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$. If $(i, j) \in \mathcal{E}$, then agent i is able to communicate with agent j. We also define the neighborhood of agent i as follows: $Nb(i) := \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\ or\ j = i\}$. Without loss of generality, we assume the graph $\mathcal{G}$ is connected

and undirected. The N agents jointly solve the following consensus optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\xi_i\sim\mathcal{D}_i}[F^i(\mathbf{x};\xi_i)], \tag{2}$$

where $f^i(\mathbf{x}) := \mathbb{E}_{\xi_i\sim\mathcal{D}_i}[F^i(\mathbf{x};\xi_i)]$ are smooth non-convex functions with different data distributions $\mathcal{D}_i$. We denote by $\mathbf{g}^i$ the mutually independent unbiased stochastic gradients sampled at points $\mathbf{x}^i \in \mathbb{R}^d$ such that $\mathbb{E}[\mathbf{g}^i] = \nabla f^i(\mathbf{x}^i)$.

In decentralized learning, consensus averaging is a critical key to maintain closeness among agents that learn to achieve the shared model parameter $\mathbf{x}^*$. Such a mechanism works well when the sampled datasets for individual agents are independent and identically distributed (IID). However, in reality, IID data is rare, and data heterogeneity needs to be considered, particularly when agents implement diverse tasks. To address this issue, numerous works have attempted to develop decentralized learning algorithms that are equipped with more complex communication protocols [3, 17, 25]. Nevertheless, they essentially fall into variants of consensus averaging and require quantification techniques to alleviate the significant communication overhead.

In this study, we turn our direction to one recently developed model merging technique that has empirically been studied for multi-task learning [1]. As introduced above, the core is to apply a suitable permutation to the weight matrices such that the new model can be adapted to unseen tasks without additional training from scratch. Thus, we extend model merging to the decentralized learning setting for the first time and mathematically show the convergence rate. We next introduce one operator that fits in the parameter space upon which vectors are defined to represent models.

Define the *stochastic model merging* operator as $\mathcal{T}(\cdot|\mathbf{\Pi}) : \mathbb{R}^d \times \mathbb{R}^d \times ... \times \mathbb{R}^d \to \mathbb{R}^d$ and the permutation matrix as $\mathbf{P}^{ij} \in \mathbb{R}^{d\times d}$ respectively, where $\mathbf{\Pi} \in \mathbb{R}^{N\times N}$ is the mixing matrix. Then we have the following:

$$\mathcal{T}(\mathbf{x}^1, ..., \mathbf{x}^q|\mathbf{\Pi}) = \sum_{j\in Nb(i)} \pi_{ij}\mathbf{P}^{ij}\mathbf{x}^j, \tag{3}$$

where $q := |Nb(i)| \leq N$, π_{ij} is the element in $\mathbf{\Pi}$ at i-th row and j-th column. It can be observed that vanilla averaging is the simplest model merging, resulting in $\mathbf{P}^{ij} = \mathbf{I}$. $\mathbf{P}^{ij}$ will be time-varying along with the update of $\mathbf{x}$ and this intuitively makes sense, as for each iteration, the model merging may apply to different features on the layers. Further details on the mixing matrix definition are provided in the supplementary material.

3.3. Algorithmic Framework

Decentralized learning typically comprises two crucial steps, communication and computation. The computation step corresponds to local model update that can possibly be accomplished by first-order methods. In this context, we combine the proposed DIMAT with SGD, momentum SGD (MSGD), and Adam [61] to develop DIMAT-SGD, DIMAT-MSGD and DIMAT-ADAM. In Algorithm 1,

Algorithm 1: DIMAT-SGD

Input : mixing matrix $\mathbf{\Pi}$, the # of iterations K, initialization $\mathbf{x}_1^i, \forall i \in \mathcal{V}$, step size α, merging frequency n

Output: $\bar{\mathbf{x}}_K = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_K^i$

1 **for** k *in* $1 : K$ **do**
2 **for** *each agent* $i \in \mathcal{V}$ **do**
3 Calculate the stochastic gradient $\mathbf{g}_k^i$;
4 **if** k *mod* n*=0* **then**
5 $\mathbf{x}_{k+1/2}^i = \sum_{j\in Nb(i)} \pi_{ij}\mathbf{P}_k^{ij}\mathbf{x}_{k+1/2}^j$;
6 **else**
7 $\mathbf{x}_{k+1/2}^i = \mathbf{x}_k^i$;
8 $\mathbf{x}_{k+1}^i = \mathbf{x}_{k+1/2}^i - \alpha\mathbf{g}_k^i$;

Line 5 implies that the frequency of merging step can be implemented periodically, which reduces the number of communication rounds. The term $k + 1/2$ denotes the update for consensus. The permutation matrix $\mathbf{P}_k^{ij}$ in model merging between two different models is essentially obtained through the activation matching [1]. Regardless of the specific detail of how permutation is completed among models, $\mathbf{P}_k^{ij}$ always remains a *doubly stochastic* matrix. Thus, to analyze the convergence property of DIMAT-SGD, we are able to couple the mixing matrix $\mathbf{\Pi}$ and permutation matrix $\mathbf{P}_k^{ij}$ in a higher dimension involving multiple agents. Analogously, when the computation step employs MSGD and Adam, the Algorithms 2 and 3 are attained accordingly. Please see the Supplementary Materials for these two algorithms and their associated analysis in more detail.

4. Main Results

4.1. Assumptions

Before presenting the main results, we first state several assumptions to serve the analysis. Throughout the rest of the analysis, the following standard assumptions hold true. We also defer all proof to the Supplementary Materials.

Assumption 1. *Problem 2 satisfies the following:*

- ***Smoothness.*** *Each function $f^i(\mathbf{x})$ is smooth with modulus L.*
- ***Bounded variances.*** *There exist $\sigma, \kappa > 0$ such that*

$$\mathbb{E}_{\xi\sim\mathcal{D}_i}[\|\nabla F^i(\mathbf{x};\xi) - \nabla f^i(\mathbf{x})\|^2] \leq \sigma^2, \forall i, \forall \mathbf{x}. \tag{4}$$

$$\frac{1}{N}\sum_{i=1}^{N}\|\nabla f^i(\mathbf{x})-\nabla f(\mathbf{x})\|^2\leq\kappa^2,\forall\mathbf{x}. \tag{5}$$

The smoothness in Assumption 1 is quite generic in decentralized learning algorithms [17, 25, 58] as it provides the guarantee of loss descent for the analysis. σ^2 signifies the upper bound of variances of stochastic gradients for local agents, while κ^2 quantifies the gradient diversity between each agent's local objective loss $f^i(\mathbf{x})$, due to the different data distributions. One can also use the bounded second moment of stochastic gradients assumption [44], which is stronger and results in a looser error bound.

Assumption 2. *The mixing matrix $\mathbf{\Pi}\in\mathbb{R}^{N\times N}$ is a symmetric doubly stochastic matrix satisfying $\lambda_1(\mathbf{\Pi})=1$ and*

$$\max\{|\lambda_2(\mathbf{\Pi})|,|\lambda_N(\mathbf{\Pi})|\}\leq\sqrt{\rho}<1, \tag{6}$$

where $0<\rho<1$, $\lambda_l(\cdot)$ is the l-th largest eigenvalue of the matrix.

The assumption for the mixing matrix $\mathbf{\Pi}$ has been utilized frequently in existing works [17, 58]. In our analysis, to consider multiple agents, we will have the expanded mixing matrix of the Kronecker product between $\mathbf{\Pi}$ and $\mathbf{I}_d$, $\mathbf{W}=\mathbf{\Pi}\otimes\mathbf{I}_d$, but the magnitudes of eigenvalues of $\mathbf{W}$ remain the same through a known result presented in the sequel and the fact that eigenvalue of $\mathbf{I}_d$ is 1.

Theorem 1. *[42] Let $\mathbf{C}\in\mathbb{R}^{N\times N}$ and $\mathbf{D}\in\mathbb{R}^{d\times d}$, with eigenvalue $\lambda\in s(\mathbf{C})$ with corresponding eigenvector $x\in\mathbb{C}^N$, and $\mu\in s(\mathbf{D})$ with corresponding eigenvector $y\in\mathbb{C}^d$, where $s(\cdot)$ signifies the spectrum of a matrix. Then $\lambda\mu$ is an eigenvalue of $\mathbf{C}\otimes\mathbf{D}$ with corresponding eigenvector $x\otimes y\in\mathbb{C}^{dN}$. Any eigenvalue of $\mathbf{C}\otimes\mathbf{D}$ arises as such a product of eigenvalues of $\mathbf{C}$ and $\mathbf{D}$.*

One immediate outcome from Theorem 1 is that $\max\{|\lambda_2(\mathbf{W})|,|\lambda_{dN}(\mathbf{W})|\}\leq\sqrt{\rho}<1$. Define the permutation matrix for all agents at the k-th time step as $\mathbf{P}_k\in\mathbb{R}^{dN\times dN}$. Based on the algorithmic frameworks, it is known that in the analysis we will have to deal with the consecutive matrix product in a form of $\prod_{\tau=1}^{k}\mathbf{W}\mathbf{P}_\tau$, which is the non-trivial part and distinguished from the existing analysis. However, $\mathbf{P}_\tau$ is a symmetric doubly stochastic matrix such that $\mathbf{W}\mathbf{P}_\tau$ is a doubly stochastic matrix. Following similarly Assumption 2, we set another assumption for $\mathbf{W}\mathbf{P}_k,\forall k\geq 1$ in the sequel.

Assumption 3. *The matrix $\mathbf{W}\mathbf{P}_k\in\mathbb{R}^{dN\times dN},\forall k\geq 1$ is a symmetric doubly stochastic matrix satisfying $\lambda_1(\mathbf{W}\mathbf{P}_k)=1$ and*

$$\max\{|\lambda_2(\mathbf{W}\mathbf{P}_k)|,|\lambda_{dN}(\mathbf{W}\mathbf{P}_k)|\}\leq\sqrt{\rho'}<1, \tag{7}$$

where $0<\rho'<1$, $\lambda_l(\cdot)$ is the l-th largest eigenvalue of the matrix.

The above assumption is mild as it only extends the similar conclusion in Assumption 2 to the matrix product. Assumption 3 not only characterizes the convergence for DIMAT, but also providing a justification of $\rho'\leq\rho$, which implies the tighter error bounds presented in the sequel. This holds due to the relationship between singular value and eigenvalue, symmetric properties, the fact that eigenvalues of a permutation matrix lie on the unit circle, and Courant–Fischer–Weyl Min-Max Principle [23]. The detailed analysis is deferred to the Supplementary Materials.

4.2. Convergence Analysis

Throughout the analysis, we define $f^*:=f(\mathbf{x}^*)>-\infty$, where $\mathbf{x}^*=\text{argmin}_{\mathbf{x}\in\mathbb{R}^d}f(\mathbf{x})$. We also set $n=1$ in the analysis.

Theorem 2. *Let Assumptions 1 and 3 hold. If the step size $\alpha\leq\min\{\frac{1-\sqrt{\rho'}}{4\sqrt{2}L},\frac{\sqrt{(1-\sqrt{\rho'})^4+64(1-\sqrt{\rho'})^2}-(1-\sqrt{\rho'})^2}{32L}\}$ in Algorithm 1, then for all $K\geq 1$, the following relationship holds true:*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}[\|\nabla f(\bar{\mathbf{x}}_k)\|^2]\leq\frac{2(f(\bar{\mathbf{x}}_0)-f^*)}{\alpha K}+\frac{4\alpha^2\sigma^2L^2}{1-\rho'}+\frac{16\alpha^2\kappa^2L^2}{(1-\sqrt{\rho'})^2}+\frac{L\alpha\sigma^2}{N}, \tag{8}$$

where $\bar{\mathbf{x}}_k=\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_k^i$.

Remark 1. *$\mathbb{E}[\|\nabla f(\bar{\mathbf{x}}_k)\|^2]$ has generically been applied as a metric to evaluate the convergence behavior in decentralized learning algorithms. In this study, the usage of stochastic model merging should have induced another metric different from $\mathbb{E}[\|\nabla f(\bar{\mathbf{x}}_k)\|^2]$. However, we still adopt it to assess the convergence rate since it only acts at the end of each iteration for all models. Within each iteration, the vanilla weight averaging has been replaced with the model merging.*

Corollary 1. *Let Assumptions 1 and 3 hold. If step size $\alpha=\mathcal{O}(\sqrt{\frac{N}{K}})$ in Algorithm 1, then for all $K\geq\max\{\frac{32NL^2}{\zeta},\frac{512NL^2}{\zeta^2+32\zeta-\zeta\sqrt{\zeta^2+64\zeta}}\}$, where $\zeta=(1-\sqrt{\rho'})^2$, we have*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}[\|\nabla f(\bar{\mathbf{x}}_k)\|^2]\leq\mathcal{O}(\sqrt{\frac{1}{NK}}+\frac{N}{(1-\rho')K}+\frac{N}{(1-\sqrt{\rho'})^2K}). \tag{9}$$

Remark 2. *Corollary 1 implies that* DIMAT-SGD *achieves the best available convergence rate but yields a*

tighter error bound due to $\rho' \leq \rho$, which suggests the impact of model merging on the topology. Before the optimization enters into the regime where $\mathcal{O}(\sqrt{\frac{1}{NK}})$ has dominated, DIMAT-SGD *reduces the optimization error, as $1-\rho'$ is larger. This intuitively tells us that the* DIMAT *framework leads to the faster initial performance gain with less communication and computation cost. On the other hand, when K is sufficiently large such that $\mathcal{O}(\sqrt{\frac{1}{NK}})$ dominates the convergence, the linear speed up due to N is achieved accordingly. In this regime, the impact of topology is much smaller, which indicates that* DIMAT-SGD *and* CDSGD *[25, 58] will have ultimately similar performance. Our empirical results will evidently support the theoretical findings. The similar theoretical implications also apply to* DIMAT-MSGD *and* DIMAT-ADAM*, given additional analysis presented in the Supplementary Materials.*

5. Experimental Results

In this section, we empirically analyze the performance of DIMAT. Our code is available here on GitHub. We compare the effectiveness of our algorithms with other baseline decentralized algorithms such as Stochastic Gradient Push (SGP) [3], consensus-based distributed SGD (CDSGD) [25] and Cross-Gradient Aggregation (CGA) [17]. However, CGA did not converge in the explored setting and has been omitted. Further exploration is warranted to conduct a meaningful comparison involving CGA. Additionally, we introduce other baselines, Weight Matching (WM), established by Ainsworth et al. [1], and Weight Averaging (WA), inspired by consensus-based distributed Adam (CDAdam) [36]. These baselines share the same setup as DIMAT but, instead of relying on activations, focus on inspecting the weights of the agents and averaging the weights of the agents, respectively. The goal in WM is to maximize the dot product between the vectorized weights of agents, formulated as a sum of bilinear assignments problem. We include these additional baselines to have a fair and direct comparison of the merging method used within the DIMAT framework. It is worth noting that WM has not been implemented for ResNet in the existing literature, and such an implementation is nontrivial. Consequently, we do not provide comparison results for WM in ResNet.

Experimental Setup: Our experimental scope covered datasets such as CIFAR-100, CIFAR-10, and Tiny ImageNet. CIFAR10 and Tiny ImageNet results can be found in the Supplementary Materials. We assessed system scalability by varying the number of agents—examining scenarios with 5 to 10, 20, and 40 agents. To evaluate the algorithm's robustness across diverse network architectures, we employed three distinct model architectures: VGG16 [60], ResNet-20 [20], and ResNet50 [20]. ResNet50 results can be found in the Supplementary Materials. Throughout the experiments, we initiated with pre-trained agents, each agent having undergone individual pre-training on their respective datasets for 100 epochs. All experiments were conducted five times for an average of 100 iterations to ensure result reliability. Each iteration includes 2 training epochs, where the agents are trained on their respective datasets, and a merge operation.

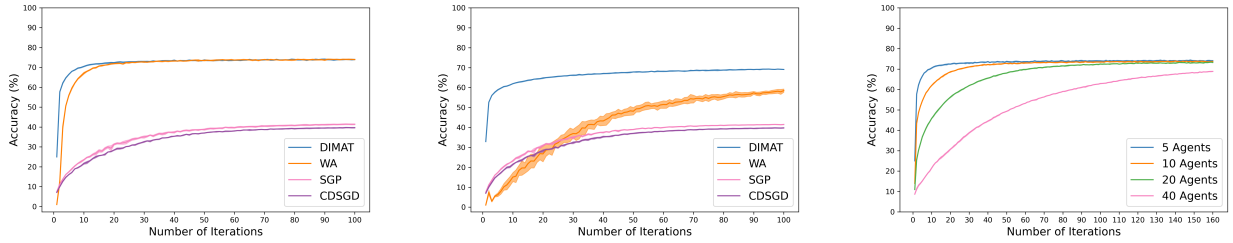
Communication and Computational Overhead: DIMAT notably requires less communication and computation than traditional decentralized learning methods. DIMAT has a merge operation once every 2 training epochs, and as such, has $0.5 * (n-1)$ communication rounds per epoch for fully connected topology and $0.5 * 2$ communication rounds per epoch for ring topology, where n is the number of agents. On the other hand, methods such as CDAdam and SGP have communication steps every mini-batch, which results in significantly more communication rounds per epoch than DIMAT. Further visual comparisons of communication are available in the supplementary material, as well as the computational results.

To evaluate DIMAT's performance, we conducted experiments comparing results under both IID and non-IID data distributions. This assessment provides insights into the algorithm's adaptability to varying data characteristics. In the IID setting, data is uniformly distributed among agents, with each agent responsible for a subset of data samples, ensuring balanced class representation. For example, with 5 agents, each handles 1/5 of the data for every class. Conversely, in the non-IID setup, significant class imbalances occur, with each agent possessing data from a smaller subset of classes. In a scenario with 100 classes and 5 agents, each agent is allocated data from 20 distinct classes, potentially creating a more challenging learning environment due to the presence of unseen data and classes.

Table 2. Comparison of test accuracy (mean±std) on CIFAR-100 with ResNet-20 architecture for 5 agents under both IID and non-IID data distribution, considering fully connected (FC) and ring topologies.

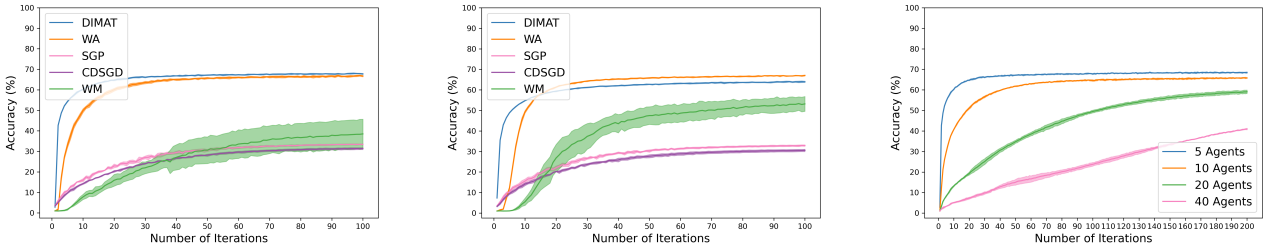
Algorithm	**IID**		**non-IID**	
	FC	Ring	FC	Ring
SGP	41.39±0.24	41.48±0.14	12.85±0.14	11.95±0.33
CDSGD	39.69±0.20	39.77±0.14	12.31±0.19	10.51±0.38
WA	73.97±0.12	58.49±0.81	**62.7±0.44**	8.00±0.51
DIMAT (ours)	**73.99±0.18**	**69.07±0.01**	62.05±0.64	**20.45±0.65**

The results of our experiments are presented in Table 2 for ResNet-20 and Table 3 for VGG16. These tables provide a comprehensive overview of testing accuracy comparisons across various network topologies under both IID and non-IID data distributions. Subsequent subsections will delve into detailed discussions regarding IID and non-IID scenarios.



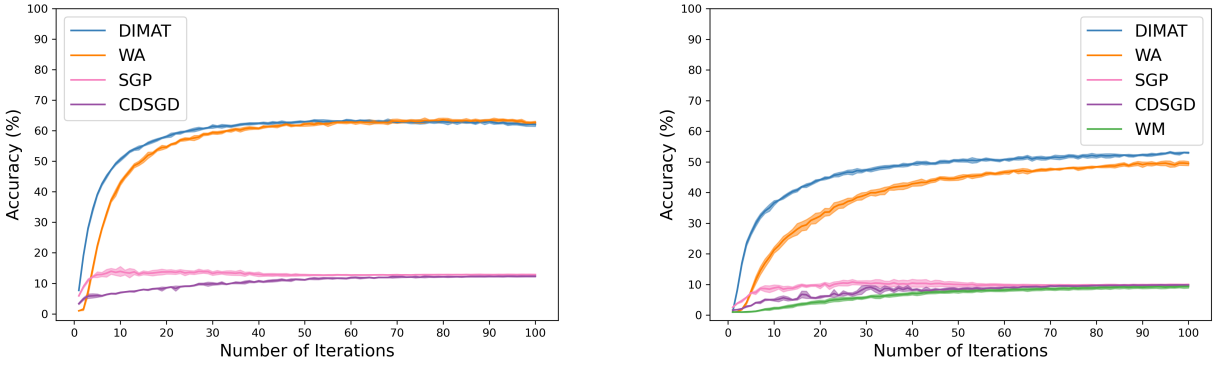
(a) FC Topology for 5 Agents (b) Ring Topology for 5 Agents (c) Scalability Analysis for FC Topology

Figure 2. Comparing algorithmic accuracy (mean±std) in fully connected (FC) (a) and ring (b) topologies with ResNet-20 architecture on CIFAR-100 IID data. The scalability with ResNet-20 architecture on CIFAR-100 IID data and fully connected topology is shown in (c).



(a) FC Topology for 5 Agents (b) Ring Topology for 5 Agents (c) Scalability Analysis for FC Topology

Figure 3. Comparing algorithmic accuracy (mean±std) in fully connected (FC) (a) and ring (b) topologies with VGG16 architecture on CIFAR-100 IID data. The scalability with VGG16 architecture on CIFAR-100 IID data and fully connected topology is shown in (c).

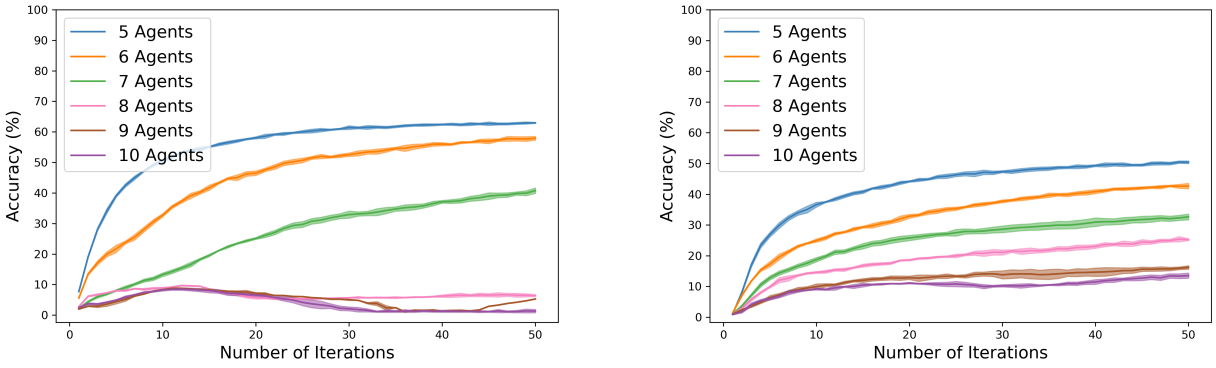


(a) ResNet-20 (b) VGG16

Figure 4. Comparing algorithmic accuracy (mean±std) in fully connected topology with ResNet-20 (a) and VGG16 (b) architecture on CIFAR-100 non-IID data for 5 agents.

5.1. Comparison of Algorithms in IID Setting

In the context of the IID setting, we conducted a thorough analysis of the results presented in Tables 2 (ResNet-20) and 3 (VGG16). Our analysis reveals important insights into the performance of the different algorithms. Visual analysis in Fig. 2 and Fig. 3 highlights DIMAT's rapid convergence across ResNet-20 and VGG16 in various topologies. This suggests DIMAT swiftly achieves accuracy stability during training, supporting our theoretical claims. DIMAT consistently outperforms or matches alternative algorithms in accuracy metrics, with Weight Averaging (WA) eventually converging similarly. Scalability analysis, depicted in Fig. 2c and Fig. 3c, indicates DIMAT maintains high accuracy and convergence as agent count increases, showcasing its robustness across architectures and topologies.



(a) ResNet-20 (b) VGG16

Figure 5. Scalability Analysis for DIMAT: Evaluating accuracy (mean±std) in fully connected topology with ResNet-20 (a) and VGG16 (b) on CIFAR-100 non-IID data across 5 to 10 agents.

Table 3. Comparison of test accuracy (mean±std) on CIFAR-100 with VGG16 architecture for 5 agents under both IID and non-IID data distribution, considering fully connected (FC) and ring topologies.

Algorithm	**IID**		**non-IID**	
	FC	Ring	FC	Ring
SGP	33.42±0.25	33.42±0.25	10.00±0.17	10.00±0.16
CDSGD	31.37±0.47	30.53±0.58	9.81±0.15	9.53±0.10
WM	38.13±6.67	52.58±2.34	9.37±0.54	13.3±0.54
WA	66.70±0.34	**67.02±0.24**	49.48±0.68	25.15±0.27
DIMAT (ours)	**68.08±0.24**	63.76±0.30	**52.99±0.24**	**25.28±0.13**

5.2. Comparison of Algorithms in Non-IID Setting

In the non-IID setting, our analysis highlights DIMAT's consistent and swift convergence compared to alternative algorithms, particularly with fully connected topology (see Fig. 4). Although DIMAT converges faster, Weight Averaging (WA) eventually achieves similar performance levels, indicating its potential effectiveness given sufficient training time. Scalability is a critical factor in evaluating the practicality of algorithms. We analyze scalability breakpoints, as shown in Fig. 5, essential for informed decision-making in real-world deployments. Understanding how algorithms perform with increasing agents offers valuable insights, shedding light on the balance between individual learning and potential biases. Additionally, our supplementary material delves deeper into this issue, addressing how agents' learning diminishes with more agents due to reduced dataset exposure and how leveraging larger, diverse datasets can mitigate biases.

6. Conclusions

This paper presents a novel decentralized deep learning framework DIMAT by leveraging the recently developed model merging techniques. DIMAT has provably shown the sublinear convergence rate and led to the tighter error bounds, compared to popular existing approaches. The comprehensive empirical studies show the superiority of DIMAT over baselines and faster model performance gain in different settings with lower communication overhead, under both IID and non-IID data distributions, which supports well the theoretical claims. Therefore, DIMAT offers a more practical option for quickly updating (potentially large) pre-trained models with (local) new data in a collaborative manner. Beyond the current work, a few future research directions include: a) exploring different permutation algorithms for model merging in decentralized learning; b) combining quantization techniques with model merging to further reduce communication bits; c) resolving scalability issues of model merging, especially with non-IID data distributions.

7. Acknowledgements

This work was partly supported by the National Science Foundation, USA under grants CAREER CNS-1845969, CPS Frontier CNS-1954556, CAREER CIF-2005804, SATC 2154119, and NSF/USDA-NIFA award number, 2021-67021-35329.

References

[1] Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022. 2, 3, 4, 6, 10

[2] Sai Aparna Aketi, Sangamesh Kodge, and Kaushik Roy. Neighborhood gradient clustering: An efficient decentralized learning method for non-iid data distributions. *arXiv preprint arXiv:2209.14390*, 2022. 3

[3] Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and

Michael Rabbat. Stochastic gradient push for distributed deep learning, 2019. 2, 4, 6

[4] Aditya Balu, Zhanhong Jiang, Sin Yong Tan, Chinmay Hegde, Young M Lee, and Soumik Sarkar. Decentralized deep learning using momentum-accelerated consensus. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3675–3679. IEEE, 2021. 2

[5] Dominique Beaini, Shenyang Huang, Joao Alex Cunha, Gabriela Moisescu-Pareja, Oleksandr Dymov, Samuel Maddrell-Mander, Callum McLean, Frederik Wenkel, Luis Müller, Jama Hussein Mohamud, et al. Towards foundational models for molecular learning on large-scale multi-task datasets. *arXiv preprint arXiv:2310.04292*, 2023. 1

[6] Enrique Tomás Martínez Beltrán, Mario Quiles Pérez, Pedro Miguel Sánchez Sánchez, Sergio López Bernal, Géróme Bovet, Manuel Gil Pérez, Gregorio Martínez Pérez, and Alberto Huertas Celdrán. Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. *IEEE Communications Surveys & Tutorials*, 2023. 1

[7] Albert S Berahas, Raghu Bollapragada, Nitish Shirish Keskar, and Ermin Wei. Balancing communication and computation in distributed optimization. *IEEE Transactions on Automatic Control*, 64(8):3141–3155, 2018. 3

[8] Xuanyu Cao, Tamer Başar, Suhas Diggavi, Yonina C Eldar, Khaled B Letaief, H Vincent Poor, and Junshan Zhang. Communication-efficient distributed learning: An overview. *IEEE journal on selected areas in communications*, 2023. 2

[9] Xiangyi Chen, Belhal Karimi, Weijie Zhao, and Ping Li. On the convergence of decentralized adaptive gradient methods. In *Asian Conference on Machine Learning*, pages 217–232. PMLR, 2023. 2, 1, 6

[10] Zhikun Chen, Daofeng Li, Jinkang Zhu, and Sihai Zhang. Dacfl: Dynamic average consensus based federated learning in decentralized topology. *arXiv preprint arXiv:2111.05505*, 2021. 1, 2

[11] Cho-Chun Chiu, Xusheng Zhang, Ting He, Shiqiang Wang, and Ananthram Swami. Laplacian matrix sampling for communication-efficient decentralized learning. *IEEE Journal on Selected Areas in Communications*, 41(4):887–901, 2023. 2

[12] Adrián Csiszárik, Melinda F Kiss, Péter Kőrösi-Szabó, Márton Muntag, Gergely Papp, and Dániel Varga. Mode combinability: Exploring convex combinations of permutation aligned models. *arXiv preprint arXiv:2308.11511*, 2023. 2

[13] Bill Dally. Hardware for deep learning. In *2023 IEEE Hot Chips 35 Symposium (HCS)*, pages 1–58. IEEE Computer Society, 2023. 1

[14] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13733–13742, 2021. 2

[15] Shachar Don-Yehiya, Elad Venezian, Colin Raffel, Noam Slonim, Yoav Katz, and Leshem Choshen. Cold fusion: Collaborative descent for distributed multitask finetuning. *arXiv preprint arXiv:2212.01378*, 2022. 3

[16] Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. *arXiv preprint arXiv:2110.06296*, 2021. 3

[17] Yasaman Esfandiari, Sin Yong Tan, Zhanhong Jiang, Aditya Balu, Ethan Herron, Chinmay Hegde, and Soumik Sarkar. Cross-gradient aggregation for decentralized learning from non-iid data. In *International Conference on Machine Learning*, pages 3036–3046. PMLR, 2021. 1, 2, 3, 4, 5, 6

[18] Fateme Fotouhi, Aditya Balu, Zhanhong Jiang, Yasaman Esfandiari, Salman Jahani, and Soumik Sarkar. Dominating set model aggregation for communication-efficient decentralized deep learning. *Neural Networks*, 171:25–39, 2024. 2

[19] Salam Hamdan, Sufyan Almajali, Moussa Ayyash, Haythem Bany Salameh, and Yaser Jararweh. An intelligent edge-enabled distributed multi-task learning architecture for large-scale iot-based cyber–physical systems. *Simulation Modelling Practice and Theory*, 122:102685, 2023. 1

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 6

[21] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017. 3

[22] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022. 3

[23] Ya Sh Ilyasov and AB Muravnik. Min-max principles with nonlinear generalized rayleigh quotients for nonlinear equations. *Journal of Mathematical Sciences*, 260(6), 2022. 5, 2

[24] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. 3

[25] Zhanhong Jiang, Aditya Balu, Chinmay Hegde, and Soumik Sarkar. Collaborative deep learning in fixed topology networks. *Advances in Neural Information Processing Systems*, 30, 2017. 1, 2, 3, 4, 5, 6

[26] Zhanhong Jiang, Aditya Balu, Chinmay Hegde, and Soumik Sarkar. On consensus-optimality trade-offs in collaborative deep learning. *Frontiers in artificial intelligence*, 4:573731, 2021. 1

[27] Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*, 2022. 3

[28] Keller Jordan, Hanie Sedghi, Olga Saukh, Rahim Entezari, and Behnam Neyshabur. Repair: Renormalizing permuted activations for interpolation repair. *arXiv preprint arXiv:2211.08403*, 2022. 3

[29] Sahar Khawatmi, Ali H Sayed, and Abdelhak M Zoubir. Decentralized clustering and linking by networked agents. *IEEE Transactions on Signal Processing*, 65(13):3526–3537, 2017. 3

[30] Batiste Le Bars, Aurélien Bellet, Marc Tommasi, Erick Lavoie, and Anne-Marie Kermarrec. Refined convergence and topology learning for decentralized sgd with heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 1672–1702. PMLR, 2023. 2

[31] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent learning: Do different neural networks learn the same representations?, 2016. 3

[32] 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. 2

[33] 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. 3

[34] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 3

[35] Giorgi Nadiradze, Amirmojtaba Sabour, Peter Davies, Shigang Li, and Dan Alistarh. Asynchronous decentralized sgd with quantized and local updates. *Advances in Neural Information Processing Systems*, 34:6829–6842, 2021. 3

[36] Parvin Nazari, Davoud Ataee Tarzanagh, and George Michailidis. Dadam: A consensus-based distributed adaptive gradient method for online optimization. *IEEE Transactions on Signal Processing*, 70:6065–6079, 2022. 2, 6, 1

[37] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020. 3

[38] Fidel A Guerrero Peña, Heitor Rapela Medeiros, Thomas Dubail, Masih Aminbeidokhti, Eric Granger, and Marco Pedersoli. Re-basin via implicit sinkhorn differentiation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20237–20246, 2023. 2

[39] Shi Pu and Angelia Nedić. Distributed stochastic gradient tracking methods. *Mathematical Programming*, 187:409–457, 2021. 3

[40] Zhen Qian, Min Chen, Zhuo Sun, Fan Zhang, Qingsong Xu, Jinzhao Guo, Zhiwei Xie, and Zhixin Zhang. Multi-task deep learning for large-scale building detail extraction from high-resolution satellite imagery. *arXiv preprint arXiv:2310.18899*, 2023. 1

[41] Malte J Rasch, Charles Mackin, Manuel Le Gallo, An Chen, Andrea Fasoli, Frédéric Odermatt, Ning Li, SR Nandakumar, Pritish Narayanan, Hsinyu Tsai, et al. Hardware-aware training for large-scale and diverse deep learning inference workloads using in-memory computing-based accelerators. *Nature Communications*, 14(1):5282, 2023. 1

[42] Kathrin Schacke. On the kronecker product. *Masters thesis, University of Waterloo*, 2004. 5

[43] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. *Advances in neural information processing systems*, 30, 2017. 3

[44] Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018. 5

[45] George Stoica, Daniel Bolya, Jakob Bjorner, Taylor Hearn, and Judy Hoffman. Zipit! merging models from different tasks without training. *arXiv preprint arXiv:2305.03053*, 2023. 2

[46] Tao Sun, Dongsheng Li, and Bao Wang. Stability and generalization of decentralized stochastic gradient descent. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9756–9764, 2021. 1

[47] Yuwei Sun, Hideya Ochiai, and Hiroshi Esaki. Decentralized deep learning for multi-access edge computing: A survey on communication efficiency and trustworthiness. *IEEE Transactions on Artificial Intelligence*, 3(6):963–972, 2021. 1

[48] Yi-Lin Sung, Linjie Li, Kevin Lin, Zhe Gan, Mohit Bansal, and Lijuan Wang. An empirical study of multimodal model merging. *arXiv preprint arXiv:2304.14933*, 2023. 2

[49] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3147–3155, 2017. 2

[50] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. d^2: Decentralized training over decentralized data. In *International Conference on Machine Learning*, pages 4848–4856. PMLR, 2018. 3

[51] Thijs Vogels, Lie He, Anastasiia Koloskova, Sai Praneeth Karimireddy, Tao Lin, Sebastian U Stich, and Martin Jaggi. Relaysum for decentralized deep learning on heterogeneous data. *Advances in Neural Information Processing Systems*, 34:28004–28015, 2021. 3

[52] Johannes Von Oswald, Seijin Kobayashi, Alexander Meulemans, Christian Henning, Benjamin F Grewe, and João Sacramento. Neural networks with late-phase weights. *arXiv preprint arXiv:2007.12927*, 2020. 3

[53] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR, 2022. 3

[54] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971, 2022. 3

[55] Qiong Wu, Xu Chen, Tao Ouyang, Zhi Zhou, Xiaoxi Zhang, Shusen Yang, and Junshan Zhang. Hiflash: Communication-efficient hierarchical federated learning with adaptive staleness control and heterogeneity-aware client-edge association. *IEEE Transactions on Parallel and Distributed Systems*, 34(5):1560–1579, 2023. 2

[56] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004. 2

[57] Chung-Yiu Yau and Hoi-To Wai. Docom-sgt: Doubly compressed momentum-assisted stochastic gradient tracking algorithm for communication efficient decentralized learning. *arXiv preprint arXiv:2202.00255*, 2022. 2

[58] Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *International Conference on Machine Learning*, pages 7184–7193. PMLR, 2019. 2, 3, 5, 6

[59] 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*, pages 3029–3039, 2021. 1, 2

[60] Xiangyu Zhang, Jianhua Zou, Kaiming He, and Jian Sun. Accelerating very deep convolutional networks for classification and detection, 2015. 6

[61] Zijun Zhang. Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*, pages 1–2. Ieee, 2018. 4

[62] Andrey Zhmoginov, Mark Sandler, Nolan Miller, Gus Kristiansen, and Max Vladymyrov. Decentralized learning with multi-headed distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8053–8063, 2023. 2