AdaGK-SGD: Adaptive Global Knowledge Guided Distributed Stochastic Gradient Descent

Hangyu Ye[1], **Weiying Xie**[1*], **Yunsong Li**[1*], **Leyuan Fang**[2]

[1] State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China
[2] College of Electrical and Information Engineering, Hunan University, Changsha 410082, China
wagyr@stu.xidian.edu.cn, wyxie@xidian.edu.cn, ysli@mail.xidian.edu.cn, leyuan_fang@hnu.edu.cn

Abstract

Distributed machine learning (DML) is promising for training large models on large datasets. In DML, multiple workers collaborate on the training of neural networks, significantly reducing the time required for neural network training. The efficiency of DML is heavily influenced by communication, making it crucial to balance the trade-off between communication cost and model performance in current research. Local methods are excellent at reducing communication costs, yet face degradation in accuracy and generalizability. Indeed, global knowledge is valuable for improving performance in local methods. However, the theoretical analysis of global knowledge validity is lacking, and global knowledge can currently only be used in the global aggregation of local methods due to communication limitations and staleness. To this end, in this paper, we establish the mechanism of global knowledge guidance and propose Adaptive Global Knowledge Guided Distributed Stochastic Gradient Descent (AdaGK-SGD) to extend the guidance of global knowledge to the whole distributed training process without any additional communication. Specifically, we define the maximum lifetime of global knowledge based on the mechanism, and establish a correlation between the maximum lifetime and the validity of global knowledge to circumvent the adverse effects of global knowledge staleness. The Maximum Lifetime of Global Knowledge module of our algorithm can be applied separately to other algorithms. In addition, considering the application, we provide a straightforward and efficient strategy for achieving the maximum lifetime adaptive setting. We establish the convergence rate of AdaGK-SGD for convex and non-convex scenarios. Numerically, we find that AdaGK-SGD can significantly improve the accuracy and generalizability of distributed algorithms compared with existing methods.

Code — https://github.com/Yehangyu-XD/AdaGK-SGD

Introduction

Distributed training is an indispensable approach to training large neural network models on large datasets (Liu et al. 2019). In distributed training applications, communication costs become a bottleneck due to the increasing depth and width of machine learning models. To address the bottleneck, periodic averaging has emerged as a promising strategy in this regard, known as local methods such as Local SGD (McMahan et al. 2017). For example, in Local SGD, each worker independently updates its local model and periodically aggregates it with other workers to update the global model, effectively reducing communication costs in distributed training. However, these methods introduce additional noise into the training process, resulting in a decline in accuracy and generalization after an equal number of iterations compared to plain distributed training (Zinkevich et al. 2010).

To tackle this issue, designing distributed machine learning (DML) algorithms using global knowledge is an effective approach for contemporary research. For example, Elastic Averaging SGD (EASGD) (Zhang, Choromanska, and LeCun 2015) verifies that linking computational parameters of workers to global variables stored in the master worker can be effective in improving convergence. Cooperative SGD (Wang and Joshi 2021) combines EASGD with local methods by creating a unified framework. SlowMo (Wang et al. 2019) solves the heterogeneity problem of local momentum buffers by using global slow momentum and achieves a communication-efficient training algorithm with momentum to improve the accuracy of the model. In DML, global knowledge refers to parameters, models or other information that are shared across a distributed system, similar to the concept of global variables in EASGD and global slow momentum in SlowMo.

Thus, algorithms with global knowledge enjoy several advantages: *1) Global knowledge improves the ability of the optimizer to escape from local saddle points.* The concept of a saddle point in mathematics refers to a stationary point of a function that is neither an extreme nor an inflection point. Due to the near-zero gradient around the saddle point, the plain SGD algorithm is prone to getting trapped and unable to escape. *2) Global knowledge suppresses the overfitting of local models.* In distributed training, local models are prone to overfitting due to data imbalance, model complexity, and local update strategies, especially in federated learning where the data is distributed non-independently and non-identically (non-i.i.d). *3) Global knowledge enhances the convergence rate of the model.* Instead of global averaging every τ local updates, global knowledge produces a proximal term to the objective function, which allows some slack between models(Wang and Joshi 2021), (Zhang, Choromanska, and LeCun 2015). This is inspired by the alternating direction method

*Correspond author.

of multipliers (ADMM) (Parikh, Boyd et al. 2014), (Boyd et al. 2011). In theory, the analysis in Parle(Chaudhari et al. 2017), EASGD (Zhang, Choromanska, and LeCun 2015) and Cooperative SGD (Wang and Joshi 2021) demonstrated the effectiveness of it on convergence rate improvement.

In this paper, we consider a crucial issue about global knowledge: Global knowledge needs to be constantly updated during the training process, but this results in significant additional communication costs. Therefore, it is crucial to ask *can we extend the global knowledge guidance to the whole process of distributed training with low communication costs?*

More Related Works and the Novelty

Distributed Machine Learning. There has been a surge of interest in DML among researchers. DML algorithms can be traced back to the work of Tsitsiklis *et al.* (Tsitsiklis, Bertsekas, and Athans 1986), which first gives upper and lower bounds on the communication complexity of distributed convex optimization in the two-worker case. Delayed Stochastic Gradient Descent (Langford, Smola, and Zinkevich 2009) improves the general optimization algorithm to achieve distributed training. It's gradually replaced by later works (Chu et al. 2006),(Mcdonald et al. 2009),(Teo et al. 2010) due to its inability to adapt to the prevailing communication rules for large-scale machine learning. However, these distributed algorithms fail to perform satisfactorily because the computational rules and communication methods limit them. Bottou *et al.* (Bottou 2010) use error decomposition to demonstrate that stochastic optimization algorithms are more effective in large-scale machine learning with computational time constraints. Parallel SGD (Zinkevich et al. 2010) lifts the computational latency constraint and ensures the effectiveness of distributed training acceleration. Wu *et al.* (Wu, Zhang, and Wang 2019) provide a theoretical analysis of the generalization stability of DML algorithms in the presence of big data.

Communication-efficient Methods. With the exponential increase in data volume and the growing complexity of neural network structures, plain DML algorithms incur huge communication costs. Mini-batch SGD (Dekel et al. 2012) converts many serial gradient-based online prediction algorithms into distributed algorithms, reducing the communication during distributed training. Local SGD (McMahan et al. 2017) performs a complete local gradient descent at each worker, breaking the communication bottleneck and significantly decreasing the communication frequency. Lin *et al.* (Lin et al. 2018) demonstrated that Local SGD outperforms Mini-batch SGD in both communication-constrained and generalizability-constrained cases. CSGD (Li et al. 2021) optimizes communication costs by enabling nodes to participate in global aggregation only when they possess a certain amount of information. While communication-efficient methods can significantly save communication costs in distributed training, they often lead to performance degradation. GT-VR (Jiang et al. 2022) is a distributed stochastic method that incorporates variance reduction techniques to mitigate the additional variance introduced during distributed training. Momentum has been proven to improve performance in non-distributed training. PR-SGD-Momentum (Yu, Jin, and Yang 2019) demonstrated that gradient descent with simultaneous momentum and parameters can improve the performance of the model. BMUF (Chen and Huo 2016) and the serial Lookahead optimizer (Zhang et al. 2019) also update the model parameters in momentum instead of the linear search method. QHM (Ma and Yarats 2018) is a quasi-hyperbolic momentum method averaging a plain SGD step with a momentum step. In the setting of smooth non-convex functions, Gitman *et al.* (Gitman et al. 2019) established the stability and asymptotic convergence results of QHM (Ma and Yarats 2018). SlowMo (Wang et al. 2019) is a framework for optimization algorithms considering momentum for distributed algorithms (Chen and Huo 2016),(Zhang et al. 2019), (Ma and Yarats 2018) which introduces global momentum to improve the performance of distributed optimization algorithms.

In contrast to prior research, this paper specifically focuses on the crucial role of global knowledge in DML and explores its fundamental characteristics at a theoretical level. Various techniques can be integrated into our framework to improve its communication efficiency. Although other methods, such as communication topology design (Huang et al. 2021), decentralization (Sun, Li, and Wang 2022) and gradient compression (Wei et al. 2022) (Abrahamyan et al. 2021) are not considered in this paper, these orthogonal techniques can definitely be applied in addition to our framework.

Contributions

We propose a novel DML algorithm, namely Adaptive Global Knowledge Guided Distributed Stochastic Gradient Descent (AdaGK-SGD), which extends the global knowledge guidance to the whole distributed training process to improve the accuracy and generalization without incurring additional communication costs. Our contributions in this paper are elaborated below in threefold.

- Methodologically, AdaGK-SGD is not only a mere algorithm, but its Maximum Lifetime of Global Knowledge (MLGK) module is able to support all existing local methods and their variants with global knowledge. In particular, drawing inspiration from the Time-to-Live (TTL) concept in communication technology (Basu et al. 2018), we aim to overcome the global knowledge staleness constraint and extend the effective interval of global knowledge beyond just during global averaging to encompass the entire distributed training process.

- Theoretically, we examine the soundness of AdaGK-SGD. Theoretical analysis poses two major challenges: 1) Establishing the mechanism of global knowledge guidance requires meticulous modeling of the distributed training process, both with and without the participation of global knowledge. 2) Modeling the correlation between the maximum lifetime of global knowledge and the benefits derived from its guidance necessitates a reasonable and effective definition of it. These works have never been pursued in previous studies and have unique difficulties for our analysis. The AdaGK-SGD algorithm achieves conver-

gence at

$$O\left(\frac{\sigma}{\sqrt{NT}}+\frac{\rho^{\frac{1}{3}}\tau^{\frac{1}{3}}\left(\sigma^{\frac{2}{3}}+h^{\frac{2}{3}}\right)}{T^{\frac{2}{3}}}+\frac{\tau}{T}\right), \quad (1)$$

in both smooth convex and non-convex cases, where N represents the network scale, T denotes the total number of iterations, σ^2 reflects gradient noise, h measures data heterogeneity, τ indicates global averaging period, and ρ is associated with data heterogeneity.

- Empirically, we validate the effectiveness of the AdaGK-SGD algorithm on various datasets and network structures. In addition, we improve the state-of-the-art algorithm using the Maximum Lifetime of Global Knowledge module and compare it with the original algorithm. The accuracy of AdaGK-SGD is significantly improved on various models and datasets, which proves its effectiveness and generalizability.

Preliminaries

We use lowercase and lowercase boldface letters to represent scalars and vectors, respectively, while matrices are denoted by uppercase boldface letters. For a vector $\boldsymbol{x}$, we denote its l_2-norm paradigm as $\|\boldsymbol{x}\|$. For a matrix $\boldsymbol{A}$, its transpose is denoted as $\boldsymbol{A}^T$, and its $Frobenius-norm$ is denoted as $\|\boldsymbol{A}\|_F$. Given two sequences $\{a_n\}$ and $\{b_n\}$, we write $a_n=O(b_n)$ if there exists a positive constant $0<C<+\infty$ such that $a_n<Cb_n$. For a function $f(x):\mathbb{R}^d\to\mathbb{R}$, we denote its gradient by $\nabla f(x)$. We use $\mathbb{E}[\cdot]$ to denote the expectation of the underlying probability space.

Problem Definition and Assumptions

DML is to solve the following problem

$$\min_{\boldsymbol{w}\in\mathbb{R}^d}\frac{1}{N}\sum_{i=1}^{N}\left[l_i(\boldsymbol{w}):=E_{\boldsymbol{\xi}_i\sim\boldsymbol{D}_i}\mathcal{L}_i\left(\boldsymbol{\xi}_i,\boldsymbol{w}\right)\right], \quad (2)$$

where $\boldsymbol{w}$ denotes weights, and $\boldsymbol{\xi}_i$ denotes the local data sampled at worker i that follows distribution $\boldsymbol{D}_i$. The expectation of the loss function $\mathcal{L}_i\left(\boldsymbol{\xi}_i,\boldsymbol{w}\right)$ of worker i is denoted by l_i, and $\boldsymbol{w}^*_{\text{Global}}$ is the solution to Equation 2. In the local methods, local models are averaged after every τ iterations. It is mathematically described as

$$\boldsymbol{w}_i^{(k+1)}=\begin{cases}\frac{1}{N}\sum_{j=1}^{N}\left(\boldsymbol{w}_j^{(k)}-\eta\boldsymbol{g}_j^{(k)}\right), & k \bmod \tau=0\\ \boldsymbol{w}_i^{(k)}-\eta\boldsymbol{g}_i^{(k)}, & \text{else}\end{cases} \quad (3)$$

where η is the learning rate, and $\boldsymbol{g}_i^{(k)}=\nabla\mathcal{L}_i\left(\boldsymbol{\xi}_i{}^{(k+1)},\boldsymbol{w}_i^{(k)}\right)$ is the stochastic gradient.

The reasonableness of the assumptions as well as the analysis regarding the actual problem situation are presented in the Supplementary Materia.

Assumption 1 (L-Lipschitz continuous (Li et al. 2020), (Reddi et al. 2021), (Chen et al. 2021), (Koloskova et al. 2020)). *The local objective function l_i is differentiable, and there exists a constant L, for each $\boldsymbol{w}_1,\boldsymbol{w}_2\in\mathbb{R}^d$:*

Algorithm 1: AdaGK-SGD

Require: network scale N, global averaging period τ, total number of iterations T, learning rate η, auxiliary variable parameter αinitial parameter $\boldsymbol{w}_{\text{init}}$.

Initialize: $\boldsymbol{w}^{(0)}\leftarrow\boldsymbol{w}_{\text{init}}$, $\boldsymbol{w}^{(0)}_{\text{Global}}\leftarrow\boldsymbol{w}_{\text{init}}$, $\boldsymbol{\mathcal{Z}}^{(0)}\leftarrow\boldsymbol{0}$, $\mathcal{G}^{(0)}\leftarrow\boldsymbol{0}$, $\mathcal{M}\leftarrow\tau$.

1: **for** $k=1,2,...,T$ every worker i **do**
2: Sample $\xi_i^{(k+1)}$, update $g_i^{(k)}=\nabla\mathcal{L}_i\left(\xi_i{}^{(k+1)},\boldsymbol{w}_i^{(k)}\right)$.
3: $\mu(k)=\max\{s:s\le k \text{ and } s \bmod \tau=0\}$.
4: **if** $k=\mu(k)$ **then**
5: $\boldsymbol{w}^{(\mu(k))}_{\text{Global}}=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{w}_i^{(\mu(k))}$.
6: Compute $\boldsymbol{\mathcal{Z}}_i^{(\mu(k))}$ base on Equation 13 or other methods.
7: **end if**
8: $\boldsymbol{w}_i^{(k+\frac{1}{2})}=\boldsymbol{w}_i^{(k)}-\eta\,\boldsymbol{g}_i^{(k)}$
9: Determine ψ based on Equation 14.
10: $\boldsymbol{\mathcal{G}}_i^{(k)}=\psi\alpha\left(\boldsymbol{\mathcal{Z}}_i^{(\mu(k))}-\boldsymbol{w}_i^{(k+\frac{1}{2})}\right)$.
11: Compute $\mathcal{M}$ of global knowledge based on Equations 19 or 22.
12: $\boldsymbol{w}_i^{(k+1)}=\boldsymbol{w}_i^{(k+\frac{1}{2})}+\boldsymbol{\mathcal{G}}_i^{(k)}$
13: **end for**

$$\|\nabla l_i\left(\boldsymbol{w}_1\right)-\nabla l_i\left(\boldsymbol{w}_2\right)\|\le L\left\|\boldsymbol{w}_1-\boldsymbol{w}_2\right\|. \quad (4)$$

Assumption 2 (Bounded gradient noise (Li et al. 2020), (Reddi et al. 2021), (Chen et al. 2021), (Koloskova et al. 2020)). *There exists a bounded positive constant σ^2, for any k and i:*

$$\mathbb{E}\left[\nabla\mathcal{L}_i\left(\boldsymbol{\xi}_i^{(k+1)},\boldsymbol{w}_i^{(k)}\right)-\nabla l_i\left(\boldsymbol{w}_i^{(k)}\right)\middle|F^{(k-1)}\right]=0, \quad (5)$$

$$\mathbb{E}\left[\left\|\nabla\mathcal{L}_i\left(\boldsymbol{\xi}_i^{(k+1)},\boldsymbol{w}_i^{(k)}\right)-\nabla l_i\left(\boldsymbol{w}_i^{(k)}\right)\right\|^2\middle|F^{(k-1)}\right]\le\sigma^2. \quad (6)$$

The priori filtration is defined as $F^{(k)}=\left\{\left\{\boldsymbol{\xi}_i^{(k)}\right\}_{i=1}^{N},\left\{\boldsymbol{w}_i^{(k)}\right\}_{i=1}^{N},\ldots,\left\{\boldsymbol{\xi}_i^{(0)}\right\}_{i=1}^{N},\left\{\boldsymbol{w}_i^{(0)}\right\}_{i=1}^{N}\right\}$.

Assumption 3 (Convexity and data heterogeneity (Chen et al. 2021), (Koloskova et al. 2020), (Sun, Li, and Wang 2022)). *In case that the objective function is convex, $h^2=\frac{1}{N}\sum_{i=1}^{N}\left\|\nabla l_i\left(\boldsymbol{w}^*\right)\right\|^2$ denotes the heterogeneity of data.*

Assumption 4 (Non-convex and data heterogeneity (Chen et al. 2021), (Koloskova et al. 2020), (Sun, Li, and Wang 2022)). *In case that the objective function is non-convex, there exists $\hat{h}>0$ such that $\frac{1}{N}\sum_{i=1}^{N}\left\|\nabla l_i(\boldsymbol{w})-\nabla l(\boldsymbol{w})\right\|^2\le\hat{h}^2$ for any $\boldsymbol{w}\in\mathbb{R}^d$.*

Adaptive Global Knowledge Guided Distributed SGD

We illustrate AdaGK-SGD in Fig. 2 and summarize it in Algorithm 1.

$\mathcal{D}$-Distributed Training Strategy

The element $\mathcal{D}$ is the backbone of distributed training, which can employ various local methods such as Local SGD, Fe-

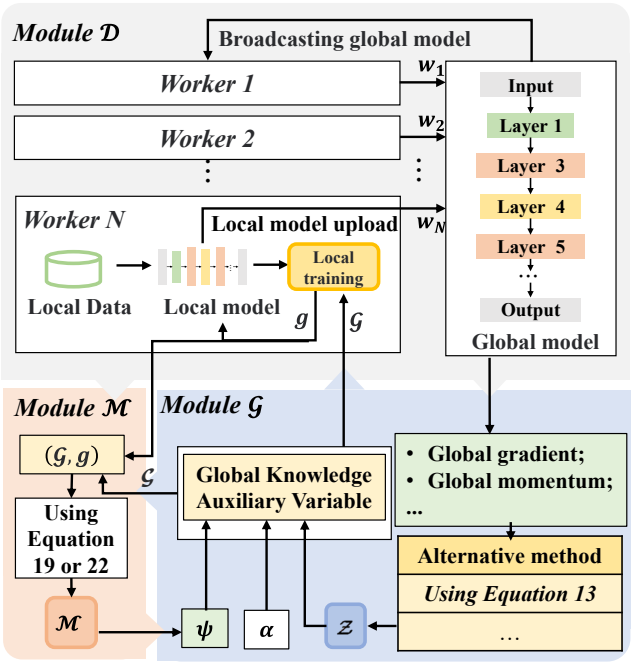

Figure 1: AdaGK-SGD consists of modules $\mathcal{D}$, $\mathcal{G}$ and $\mathcal{M}$. Module $\mathcal{D}$ supports a variety of basic DML algorithms, such as Local SGD, FedAvg, Cooperative SGD, and so on. Module $\mathcal{G}$ updates the global knowledge according to module $\mathcal{D}$, and provides continuous global knowledge guidance for the local training of each worker in module $\mathcal{D}$. Here, $\mathcal{Z}$ represents the auxiliary variable correction, which can be calculated by various methods. α indicates the weight of the auxiliary variable correction. ψ obtained from module $\mathcal{M}$ is a parameter that determines whether module $\mathcal{G}$ is enabled. Module $\mathcal{M}$ can adaptively calculate the maximum lifetime of global knowledge based on the information provided by module $\mathcal{D}$ and module $\mathcal{G}$.

dAvg, and Cooperative SGD. To demonstrate the performance, AdaGK-SGD employs the simplest form of Local SGD as its $\mathcal{D}$. First, the local model is initialized to the current global model

$$\boldsymbol{w}_i^{(\mu(k))} = \boldsymbol{w}_{\text{Global}}^{(\mu(k))}, \tag{7}$$

where $\mu(k) = \max\{s : s \leq k \text{ and } s \bmod \tau = 0\}$. Then each worker calculates its own local gradient and updates the local model. However, unlike the ordinary Local SGD, the update here introduces a global knowledge correction term $\mathcal{G}_i^{(\mu(k))}$, which is the key to performance improvement:

$$\boldsymbol{g}_i^{(\mu(k)+t)} = \nabla \mathcal{L}_i \left(\boldsymbol{\xi}_i^{(\mu(k)+t+1)}, \boldsymbol{w}_i^{(\mu(k)+t)} \right), \tag{8}$$

$$\boldsymbol{w}_i^{(\mu(k)+t+\frac{1}{2})} = \boldsymbol{w}_i^{(\mu(k)+t)} - \eta\, \boldsymbol{g}_i^{(\mu(k)+t)}, \tag{9}$$

$$\boldsymbol{w}_i^{(\mu(k)+t+1)} = \boldsymbol{w}_i^{(\mu(k)+t+\frac{1}{2})} + \boldsymbol{\mathcal{G}}_i^{(\mu(k))}, \tag{10}$$

where $0 < t \leq \tau$.

Finally, when τ local updates are performed, all workers aggregate the latest local model to update the global model:

$$\boldsymbol{w}_{\text{Global}}^{(\mu(k)+\tau)} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{w}_i^{(\mu(k)+\tau)}. \tag{11}$$

$\mathcal{G}$-Global Knowledge Auxiliary Variable

As shown in Equation 10, this module is denoted by $\mathcal{G}_i^{(\mu(k))}$ and acts on the local training. Its explicit expression is

$$\boldsymbol{\mathcal{G}}_i^{(\mu(k))} = \psi \alpha \left(\boldsymbol{\mathcal{Z}}_i^{(\mu(k))} - \boldsymbol{w}_i^{(\mu(k)+t+\frac{1}{2})} \right), \tag{12}$$

where ψ is the enabling parameter determined by the maximum lifetime of global knowledge, α denotes the weight of the auxiliary variable correction, and $\boldsymbol{\mathcal{Z}}_i^{(\mu(k))}$ denotes the true impact of global knowledge. In AdaGK-SGD, $\boldsymbol{\mathcal{Z}}_i^{(\mu(k))}$ is obtained by:

$$\boldsymbol{\mathcal{Z}}_i^{(\mu(k))} = \frac{N}{N-1} \left(\boldsymbol{w}_{\text{Global}}^{(\mu(k))} - \frac{1}{N} \boldsymbol{w}_i^{(\mu(\mu(k)-1)+\tau)} \right). \tag{13}$$

The purpose of subtracting $\frac{1}{N} \boldsymbol{w}_i^{(\mu(\mu(k)-1)+\tau)}$ here is to eliminate known outdated information, while the purpose of multiplying by $\frac{N}{N-1}$ is to align the scales of the scale correction terms of the model weights to make it easier to set the weights of the correction terms.

$\mathcal{M}$-Maximum Lifetime of Global Knowledge (MLGK)

According to the analysis about mechanisms of global knowledge guidance, the benefit of global knowledge lies in the adjustment of the initial point of the local update. However, during the process of distributed training, global knowledge continuously becomes stale and loses its usefulness or even adversely affects the training. In the field of Telecommunication Engineering, a similar problem is faced: time-sensitive objects such as Routing Information Base (RIB) and messages can adversely affect system performance because of becoming stale, but updating them results in a non-negligible communication cost. A fairly effective way to achieve an optimal trade-off between communication costs and system performance is to set a Time-to-Live (TTL) for such objects, which indicates their maximum lifetime(Basu et al. 2018). It is worth noting that global knowledge also has similar characteristics. Inspired by this, we give the definition of the maximum lifetime of global knowledge in distributed training:

Definition 1 *If the global knowledge is updated in the $t_0 - th$ iteration and local training can no longer profit from exploiting the global knowledge in the $t_1 - th$ iteration, the maximum lifetime of the global knowledge is $t_1 - t_0$.*

Once the global knowledge exceeds its maximum lifetime, it is no longer involved in local updates to prevent the adverse effects on training caused by staleness. This is controlled by the following parameters:

$$\psi = \begin{cases} 0, & t > \mathcal{M} \\ 1. & \text{else} \end{cases} \tag{14}$$

Adaptive $\mathcal{M}$ Setting

In this section, we discuss the correlation between the maximum lifetime of global knowledge and its validity in both the convex (Under Assumptions 1-3) and non-convex scenarios (Under Assumptions 1,2,4), and present an implementation of the $\mathcal{M}$ adaptive setting.

Convex Scenario According to the theory of optimization and Definition 2, the necessary and sufficient condition for global knowledge to be beneficial for training is to bring the model weights closer to the global optimal solution. When the numerical space constituted by the model weights is a completely endowed linear space (Banach space), it can be expressed by

$$\begin{aligned} &\mathbb{E}\left\|\boldsymbol{w}^{*}-\left(\boldsymbol{w}_{i}^{(t)}-\boldsymbol{g}_{i}^{(t)}+\boldsymbol{\mathcal{G}}_{i}\right)\right\| \\ &\leq \mathbb{E}\left\|\boldsymbol{w}^{*}-\left(\boldsymbol{w}_{i}^{(t)}-\boldsymbol{g}_{i}^{(t)}\right)\right\|, \end{aligned} \tag{15}$$

where $\boldsymbol{w}^{*}$ denotes the optimal solution in the convex scenario, $\boldsymbol{w}_{i}^{(t)}$ denotes the initial point for the current local update, $\boldsymbol{g}_{i}^{(t)}$ denotes the gradient of the local update, and $\mathcal{G}_i$ denotes the adjustment of the global knowledge to the initial point.

Under Assumption 3, the optimal solution for the local data can be used to approximate the global optimal solution, with the learning rate set at a reasonable level. Accordingly, we can simplify Equation 15 to

$$\begin{aligned} &\mathbb{E}\left\|\boldsymbol{w}_{i}^{*}-\left(\boldsymbol{w}_{i}^{(t)}-\boldsymbol{g}_{i}^{(t)}+\boldsymbol{\mathcal{G}}_{i}\right)\right\| \\ &\leq \mathbb{E}\left\|\boldsymbol{w}_{i}^{*}-\left(\boldsymbol{w}_{i}^{(t)}-\boldsymbol{g}_{i}^{(t)}\right)\right\|, \end{aligned} \tag{16}$$

where $\boldsymbol{w}_{i}^{*}=\boldsymbol{w}_{i}^{(t)}-\sum_{k=t}^{T}\boldsymbol{g}_{i}^{(k)}$ denotes the optimal solution for the local data. According to the rules for local model updating, Equation 16 can be converted into Equation 17:

$$\mathbb{E}\left\|\sum_{k=t+1}^{T}\boldsymbol{g}_{i}^{(k)}+\boldsymbol{\mathcal{G}}_{i}\right\| \leq \mathbb{E}\left\|\sum_{k=t+1}^{T}\boldsymbol{g}_{i}^{(k)}\right\|. \tag{17}$$

Since $\boldsymbol{\mathcal{G}}_{i}$ is much smaller than $\boldsymbol{g}_{\text{local}}=\sum_{k=t+1}^{T}\boldsymbol{g}_{i}^{(k)}$, a sufficient condition for Equation 17 to hold is that the cosine similarity between $\boldsymbol{\mathcal{G}}_{i}$ and $\boldsymbol{g}_{\text{local}}$ is negative. According to Assumption 2, the angle between $\boldsymbol{g}_{i}^{(t)}$ and $\boldsymbol{g}_{\text{local}}$ has an upper bound, so the rule for adaptive $\mathcal{M}$ setting in the convex scenario is as follows.

Remark 1 *If the random noise is sufficiently small, the necessary and sufficient condition for global knowledge to be beneficial for local training in the convex scenario is*

$$\mathbb{E}\left(\cos\left(\boldsymbol{\mathcal{G}}_{i},\boldsymbol{g}_{i}^{(t)}\right)\right)\leq 0, \tag{18}$$

where $\cos\left(\boldsymbol{a},\boldsymbol{b}\right)$ *denotes the cosine similarity between vectors* $\boldsymbol{a}$ *and* $\boldsymbol{b}$*. There exists* $\mathcal{M}$ *satisfying*

$$\begin{cases}\mathbb{E}\left(\cos\left(\boldsymbol{\mathcal{G}}_{i},\boldsymbol{g}_{i}^{(t)}\right)\right)\leq 0, t\leq\mathcal{M}\\ \mathbb{E}\left(\cos\left(\boldsymbol{\mathcal{G}}_{i},\boldsymbol{g}_{i}^{(t)}\right)\right)>0, t>\mathcal{M}\end{cases} \tag{19}$$

in the convex scenario, and the maximum lifetime of global knowledge is set to $\mathcal{M}$*.*

Non-Convex Scenario In non-convex scenarios, there are multiple possible convergence points for the model weights. Data heterogeneity between different workers may lead to local models converging to far apart convergence points in distributed training. Under Assumption 2, since the local training starts with the latest global model, the difference in the convergence points of the local models can be characterized by the expectation of the average cosine similarity between the gradients of the local update at each worker as shown in Equation 20

$$Y(t,i)=\mathbb{E}\left(\frac{1}{N-1}\sum_{j=1,j\neq i}^{N}\cos\left(\boldsymbol{g}_{i}^{(t)},\boldsymbol{g}_{j}^{(t)}\right)\right). \tag{20}$$

A key effect of global knowledge is to guide the local models of individual workers to update towards the same convergence point. When $Y(t,i)$ is positive, the convergence point of worker i is similar to that of other workers. In this case, the maximum lifetime of global knowledge can be considered according to the convex scenario. Therefore, the rule for setting $\mathcal{M}$ in the non-convex scenario is as follows.

Remark 2 *If the random noise is sufficiently small, the necessary and sufficient condition for global knowledge to be beneficial for local training in the non-convex scenario is* $Y(t,i)\leq 0$ *or* $\mathbb{E}\left(\cos\left(\boldsymbol{\mathcal{G}}_{i},\boldsymbol{g}_{i}^{(t)}\right)\right)\leq 0$*. There exists* $\mathcal{M}_0$ *and* $\mathcal{M}$ *satisfying*

$$\begin{cases}Y(t,i)\leq 0, t\leq\mathcal{M}_0\\ Y(t,i)>0, t>\mathcal{M}_0\end{cases} \tag{21}$$

$$\begin{cases}\mathbb{E}\left(\cos\left(\boldsymbol{\mathcal{G}}_{i},\boldsymbol{g}_{i}^{(t)}\right)\right)\leq 0, \mathcal{M}_0<t\leq\mathcal{M}\\ \mathbb{E}\left(\cos\left(\boldsymbol{\mathcal{G}}_{i},\boldsymbol{g}_{i}^{(t)}\right)\right)>0, t>\mathcal{M}>\mathcal{M}_0\end{cases} \tag{22}$$

in the non-convex scenario, and the maximum lifetime of global knowledge is set to $\mathcal{M}$*.*

Implementation of the $\mathcal{M}$ Adaptive Setting Considering application scenarios, we give a simple and effective strategy for setting $\mathcal{M}$ based on the analysis in and because setting strictly according to theory leads to additional communication costs and a lot of complex calculations. The training loss of the model reflects to a certain extent the gap between the model and the point of convergence. Therefore, the value of $\mathcal{M}$ can be determined by identifying the duration during which the guidance of global knowledge does not result in an increase in training losses. In the non-convex case, we need to set a lower bound $\mathcal{M}_0$ for $\mathcal{M}$ based on the heterogeneity of the data to ensure $Y(t,i)\leq 0$.

The Convergence of AdaGK-SGD

In this sectiion, we analyze the convergence performance of AdaGK-SGD, and the detailed proof is in the Appendix

Convex Scenario

Theorem 1 *Under Assumptions 1-3, if*

$$\eta=\min\left\{\frac{N}{12L\tau},\left(\frac{r_0}{r_1\left(T+1\right)}\right)^{\frac{1}{2}},\left(\frac{r_0}{r_2\left(T+1\right)}\right)^{\frac{1}{3}}\right\}, \tag{23}$$

where $r_0 = 2N\mathbb{E}\left\|\bar{\boldsymbol{w}}^{(0)} - \boldsymbol{w}^*\right\|^2$, $r_1 = \frac{2\sigma^2}{N^2}$, $r_2 = \frac{36Lh^2(1+\rho\tau)}{N^2} + \frac{12L\sigma^2(1+\rho\tau)}{N^2}$ *, it holds for any* T *that:*

$$\begin{aligned}&\mathbb{E}\left[l\left(\hat{\boldsymbol{w}}^{(T)}\right) - l\left(\boldsymbol{w}^*\right)\right] \\ &= O\left(\frac{\sigma}{\sqrt{NT}} + \frac{\rho^{\frac{1}{3}}\tau^{\frac{1}{3}}\left(\sigma^{\frac{2}{3}} + h^{\frac{2}{3}}\right)}{T^{\frac{2}{3}}} + \frac{\tau}{T}\right),\end{aligned} \tag{24}$$

where $\bar{\boldsymbol{w}}^{(k)} = \frac{1}{N}\sum_{i=1}^{N}\bar{\boldsymbol{w}}_i^{(k)}$, $\hat{\boldsymbol{w}}^{(T)} = \frac{1}{T+1}\sum_{k=0}^{T}\bar{\boldsymbol{w}}^{(k)}$, $\rho \in \left(1 - \frac{2}{\tau}, 1\right)$.

Non-Convex Scenario

Theorem 2: *Under assumptions 1,2 and 4, if* η *is the same as in Equation 23 with* $r_0 = 8N\mathbb{E}\left[l\left(\bar{\boldsymbol{w}}^{(0)}\right)\right]$, $r_1 = \frac{4L\sigma^2}{N^2}$, $r_2 = \frac{72L\hat{h}^2(1+\rho\tau)}{N^2} + \frac{24L\sigma^2(1+\rho\tau)}{N^2}$ *it holds for any* T *that:*

$$\begin{aligned}&\frac{1}{T+1}\sum_{k=0}^{T}\mathbb{E}\left\|\nabla l\left(\bar{\boldsymbol{w}}^{(k)}\right)\right\|^2 \\ &= O\left(\frac{\sigma}{\sqrt{NT}} + \frac{\rho^{\frac{1}{3}}\tau^{\frac{1}{3}}\left(\sigma^{\frac{2}{3}} + \hat{h}^{\frac{2}{3}}\right)}{T^{\frac{2}{3}}} + \frac{\tau}{T}\right).\end{aligned} \tag{25}$$

Transient Stage Comparison

The transient stage refers to the finite iterations of the distributed algorithm before it reaches the linear speedup stage. In other words, when the optimization is in the transient phase, T is still relatively small so the $non - NT$ terms (such as in the second and third terms in Equation 1) dominate the convergence rate. The shorter the transient stage of a distributed algorithm is, the earlier the optimizer can enter the linear speedup stage, thus the transient stage is a crucial indicator of the generalizability of a distributed algorithm.

Ignoring the effects of σ and h, according to Theorem 1 and 2 convergence rate is determined by $\frac{\rho^{\frac{1}{3}}\tau^{\frac{1}{3}}}{T^{\frac{2}{3}}}$, $\frac{\tau}{T}$ and $\frac{1}{\sqrt{NT}}$ terms. According to the definition, the transient stage ends if the $\frac{1}{\sqrt{NT}}$ term dominates the convergence rate:

$$\begin{aligned}&\max\left\{\frac{\rho^{\frac{1}{3}}\tau^{\frac{1}{3}}}{T^{\frac{2}{3}}}, \frac{\tau}{T}\right\} \leq \frac{1}{\sqrt{NT}} \\ &\Rightarrow T_{\text{our}}^{'} \geq \max\left\{N^3\rho^2\tau^2, N\tau^2\right\}.\end{aligned} \tag{26}$$

So, the transient stage of AdaGK-SGD is

$$T_{\text{our}}^{'} = \Omega\left(N^3\rho^2\tau^2\right). \tag{27}$$

Local SGD is the representative base local method. According to (Chen et al. 2021), the convergence rate of Local SGD is determined by the $\frac{\tau^{\frac{1}{3}}}{T^{\frac{2}{3}}}$, $\frac{\tau}{T}$ and $\frac{1}{\sqrt{NT}}$ terms. The transient stage of Local SGD is

$$T_{\text{local SGD}}^{'} = \Omega\left(N^3\tau^2\right). \tag{28}$$

Since $1 - \frac{2}{\tau} < \rho < 1$, it is clear that $T_{\text{our}}^{'} < T_{\text{local SGD}}^{'}$.

Experiments

In this section, we evaluate AdaGK-SGD and the improved version with MLGK module of SlowMo (Wang et al. 2019), EASGD (Zhang, Choromanska, and LeCun 2015), and BMUF-Adam (Chen, Ding, and Huo 2020) on a variety of different image classification models and datasets. In addition, we analyze the key parameters of AdaGK-SGD. The datasets we use for our experiments are CIFAR10/100 and ILSVRC2012. All experiments are independently replicated three times and the results are subsequently averaged.

Experimental Settings

To verify the effectiveness of AdaGK-SGD in improving training performance as well as ensuring low communication cost, we set the baseline as Local SGD. The TOP-1 test accuracy of AdaGK-SGD and improved algorithms compared with that of the baseline and the original version of SlowMo, EASGD, and BMUF-Adam. The parameters specific to SlowMo, EASGD, and BMUF-Adam are set exactly as optimal in their description. All experiments on the dataset CIFAR are performed on 4 NVIDIA GTX 3090 GPUs. All experiments on the dataset ILSVRC2012 are performed on 4 NVIDIA A100-SXM. All workers form a decentralized network with Ring All-Reduce as the communication primitive. To ensure the reliability and validity of the experiments, the models in training are implemented with Pytorch. When it is not necessary to specify the parameters, the epoch is set to 100, and the local Batch Size is set to 256. All experiments use the warm-up (Goyal et al. 2017) algorithm to improve convergence, specifically the learning rate is linearly increased to 0.01 in the earliest 5 epochs and then decreases to 10^{-6} according to the cosine.

Performance Evaluation

Performance on CIFAR-10/100. In order to simulate the harsh communication conditions in real applications, we chose a relatively large global averaging period that, *i.e.*, $\tau = 20$. The value of the parameter α follows the Lemma (named Best Choice of α) presented in (Wang and Joshi 2021), *i.e.*, $\alpha = 2/(N+2)$. We present representative experimental results with ResNet in Table 1 and complete results in Appendix. Not surprisingly, AdaGK-SGD shows better performance on both datasets. Specifically with VggNet, AdaGK-SGD achieves 0.757% accuracy improvement over the baseline method on the CIFAR-10 dataset and 0.875% on the CIFAR-100 dataset and outperforms existing algorithms such as EASGD and SlowMo. By incorporating the MLGK module, significant accuracy enhancements of 1.41% and 0.5% are observed on the CIFAR-10 and CIFAR-100 datasets respectively, when compared to the EASGD algorithm alone. By utilizing the MLGK module to enhance the SlowMo algorithm, a significant increase in accuracy of 0.787% for CIFAR-10 dataset and 0.525% for CIFAR-100 dataset can be achieved. In addition, observation of the accuracy curves reveals that the convergence process of AdaGK-SGD is smoother than both EASGD and SlowMo. Since the underlying optimizer of BMUF-Adam is Adam, which is more advanced than SGD, BMUF-Adam outperforms AdaGK-SGD in the experiments on the single-branch

DATASET	METHOD	EPOCHS	ACC(%)	ΔAcc (%)
CIFAR-10	LOCAL SGD	100	87.472	-
	SLOWMO	100	88.385	-
	EASGD	100	87.897	-
	BMUF-ADAM	100	86.360	-
	ADAGK-SGD	100	**88.478**	**↑ 1.006**
	MLGK-SLOWMO	100	**88.660**	**↑ 0.275**
	MLGK-EASGD	100	**88.232**	**↑ 0.335**
	MLGK-BMUF-ADAM	100	**86.830**	**↑ 0.470**
CIFAR-100	LOCAL SGD	100	59.240	-
	SLOWMO	100	59.220	-
	EASGD	100	58.870	-
	BMUF-ADAM	100	53.600	-
	ADAGK-SGD	100	**60.710**	**↑ 1.470**
	MLGK-SLOWMO	100	**59.415**	**↑ 0.195**
	MLGK-EASGD	100	**59.310**	**↑ 0.440**
	MLGK-BMUF-ADAM	100	**54.330**	**↑ 0.730**

Table 1: TOP-1 Accuracy with ResNet

DATASET	METHOD	EPOCHS	ACC(%)	ΔAcc (%)
ILSVRC2012	LOCAL SGD	100	62.316	-
	ADAGK-SGD	**100**	**62.818**	**↑ 0.502**

Table 2: TOP-1 Accuracy on ILSVRC2012

network. However, the MLGK module of AdaGK-SGD is also effective for BMUF-Adam, achieving an accuracy improvement of 0.078% on the CIFAR-10 dataset and 0.757% on the CIFAR-100 dataset. There are consistent experimental results with other models.

Performance on ILSVRC2012. For the experiments on ILSVRC2012, we use ResNet18. The Batch Size in the experiments is set to 128. Parameters are set as experiments on CIFAR-10/100. The results of the experiment are shown in Table 2. On large-scale datasets, AdaGK-SGD is also effective, gaining a 0.502% relative accuracy improvement.

Parametric Analysis

We investigate the parameter sensitivity of AdaGK-SGD on the CIFAR-100 dataset with DenseNet40.

Performance with Different Global Averaging Periods. We now examine how AdaGK-SGD is robust to different synchronization periods. To this end, we compare the performance of AdaGK-SGD and Local SGD under a variety of global average period settings. The settings of the parameters α, learning rate η, and Batch Size are identical to the previous experiments. The global averaging period is set to $\tau \in \{3,\ 5,\ 10,\ 20,\ 40\}$. The results of the experiments are shown in Fig. 2. There is a steady performance improvement of AdaGK-SGD for all global averaging period cases. In addition, from the curve of relative performance improvement, it can be found that the performance improvement of AdaGK-SGD is more obvious as the global average period grows. When the global average period is raised from t=3 to t=40, the relative accuracy improvement is raised from 0.425% to 0.972%. In practice, the larger the global averaging period, the lower the communication cost, so a large global averaging period is often used. The experimental results show that AdaGK-SGD can be well adapted to practical application scenarios.

METHOD	BATCH SIZE				
	16	32	64	128	256
LOCAL SGD	39.107	52.787	59.462	62.797	62.522
OURS	39.842	54.637	60.070	63.722	64.417
ΔAcc (%)	**↑ 0.735**	**↑ 1.850**	**↑ 0.608**	**↑ 0.925**	**↑ 1.895**

Table 3: *Acc v.s.* Batchsizes

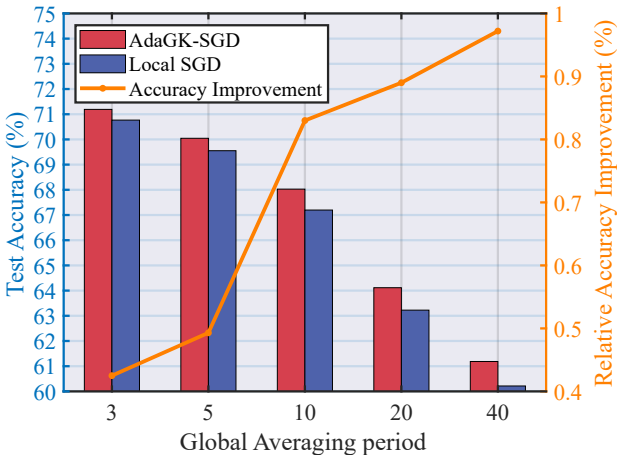

Figure 2: Comparison of Performance with different synchronization periods. The enhancement of algorithmic performance achieved by AdaGK-SGD is effective across various global average period τ settings and exhibits increasing effectiveness as τ increases.

Performance with Different Batch Sizes. We now examine how AdaGK-SGD is robust to different Batch Sizes. Table 3 displays the TOP1 test accuracy of AdaGK-SGD and Local SGD for various Batch Sizes, while maintaining consistent settings for parameters α, learning rate η, and global synchronization period τ as described in previous section. The experimental results indicate that AdaGK-SGD significantly improves algorithm performance across various Batch Sizes.

Conclusion

In this paper, we analyzed the mechanism of global knowledge guidance in DML and proposed AdaGK-SGD, the first framework to extend the guidance of global knowledge to the whole distributed training process without any additional communication. Under general assumptions, we defined, analyzed and implemented efficient settings for the maximum lifetime of global knowledge, and we established the theoretical convergence of AdaGK-SGD. We perform extensive numerical experiments to verify the efficiency of AdaGK-SGD. The numerical results show that AdaGK-SGD can significantly improve the accuracy and generalizability of distributed algorithms compared with existing methods.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 62121001, 62322117, 62371365, U24B20136, and U22B2014.

References

Abrahamyan, L.; Chen, Y.; Bekoulis, G.; and Deligiannis, N. 2021. Learned gradient compression for distributed deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12): 7330–7344.

Basu, S.; Sundarrajan, A.; Ghaderi, J.; Shakkottai, S.; and Sitaraman, R. 2018. Adaptive TTL-based caching for content delivery. *IEEE/ACM Transactions on Networking*, 26(3): 1063–1077.

Bottou, L. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of International Conference on Computational Statistics (ICCS)*.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J.; et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1): 1–122.

Chaudhari, P.; Baldassi, C.; Zecchina, R.; Soatto, S.; Talwalkar, A.; and Oberman, A. 2017. Parle: parallelizing stochastic gradient descent. *arXiv preprint arXiv:1707.00424*.

Chen, K.; Ding, H.; and Huo, Q. 2020. Parallelizing adam optimizer with blockwise model-update filtering. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Chen, K.; and Huo, Q. 2016. Scalable training of deep learning machines by incremental block training with intra-block parallel optimization and blockwise model-update filtering. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Chen, Y.; Yuan, K.; Zhang, Y.; Pan, P.; Xu, Y.; and Yin, W. 2021. Accelerating gossip SGD with periodic global averaging. In *Proceedings of International Conference on Machine Learning (ICML)*.

Chu, C.-T.; Kim, S.; Lin, Y.-A.; Yu, Y.; Bradski, G.; Olukotun, K.; and Ng, A. 2006. Map-reduce for machine learning on multicore. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 19.

Cutkosky, A.; and Busa-Fekete, R. 2018. Distributed stochastic optimization via adaptive SGD. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 31.

Dandi, Y.; Barba, L.; and Jaggi, M. 2022. Implicit gradient alignment in distributed and federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 36, 6454–6462.

Dekel, O.; Gilad-Bachrach, R.; Shamir, O.; and Xiao, L. 2012. Optimal Distributed Online Prediction Using Mini-Batches. *Journal of Machine Learning Research*, 13(1).

Ezzeldin, Y. H.; Yan, S.; He, C.; Ferrara, E.; and Avestimehr, A. S. 2023. Fairfed: Enabling group fairness in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 37, 7494–7502.

Gitman, I.; Lang, H.; Zhang, P.; and Xiao, L. 2019. Understanding the role of momentum in stochastic gradient methods. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 32.

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

Huang, J.; Majumder, P.; Kim, S.; Muzahid, A.; Yum, K. H.; and Kim, E. J. 2021. Communication algorithm-architecture co-design for distributed deep learning. In *Proceedings of International Symposium on Computer Architecture (ISCA)*.

Jiang, X.; Zeng, X.; Sun, J.; and Chen, J. 2022. Distributed stochastic gradient tracking algorithm with variance reduction for non-convex optimization. *IEEE Transactions on Neural Networks and Learning Systems*.

Koloskova, A.; Loizou, N.; Boreiri, S.; Jaggi, M.; and Stich, S. 2020. A unified theory of decentralized sgd with changing topology and local updates. In *Proceedings of International Conference on Machine Learning (ICML)*.

Langford, J.; Smola, A.; and Zinkevich, M. 2009. Slow learners are fast. *arXiv preprint arXiv:0911.0491*.

Li, L.; Xu, W.; Chen, T.; Giannakis, G. B.; and Ling, Q. 2019. RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, 1544–1551.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems (MLSys)*.

Li, W.; Wu, Z.; Chen, T.; Li, L.; and Ling, Q. 2021. Communication-censored distributed stochastic gradient descent. *IEEE Transactions on Neural Networks and Learning Systems*, 33: 6831–6843.

Lian, X.; Zhang, C.; Zhang, H.; Hsieh, C.-J.; Zhang, W.; and Liu, J. 2017. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.

Lin, T.; Stich, S. U.; Patel, K. K.; and Jaggi, M. 2018. Don't use large mini-batches, use local sgd. *arXiv preprint arXiv:1808.07217*.

Liu, F.; Wu, X.; Ge, S.; Fan, W.; and Zou, Y. 2020. Federated learning for vision-and-language grounding problems. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, 11572–11579.

Liu, M. 2024. Algorithmic Foundation of Federated Learning with Sequential Data. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, 22675–22675.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ma, J.; and Yarats, D. 2018. Quasi-hyperbolic momentum and Adam for deep learning. *arXiv preprint arXiv:1810.06801*.

Mcdonald, R.; Mohri, M.; Silberman, N.; Walker, D.; and Mann, G. 2009. Efficient large-scale distributed training of conditional maximum entropy models. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 22.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics (AISTATS)*.

Nagalapatti, L.; and Narayanam, R. 2021. Game of gradients: Mitigating irrelevant clients in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, 9046–9054.

Parikh, N.; Boyd, S.; et al. 2014. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3): 127–239.

Reddi, S.; Charles, Z.; Zaheer, M.; Garrett, Z.; Rush, K.; Konečnỳ, J.; Kumar, S.; and McMahan, H. B. 2021. Adaptive Federated Optimization. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Shen, S.; Cheng, Y.; Liu, J.; and Xu, L. 2021. STL-SGD: Speeding up local SGD with stagewise communication period. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, 9576–9584.

Sheng, V. S.; and Zhang, J. 2019. Machine learning with crowdsourcing: A brief summary of the past research and future directions. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, 9837–9843.

Siddhant, A.; Goyal, A.; and Metallinou, A. 2019. Unsupervised transfer learning for spoken language understanding in intelligent agents. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, 4959–4966.

Sun, T.; Li, D.; and Wang, B. 2022. Decentralized federated averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 4289–4301.

Teo, C. H.; Vishwanathan, S.; Smola, A.; and Le, Q. V. 2010. Bundle Methods for Regularized Risk Minimization. *Journal of Machine Learning Research*, 11(1).

Thapa, C.; Arachchige, P. C. M.; Camtepe, S.; and Sun, L. 2022. Splitfed: When federated learning meets split learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 36, 8485–8493.

Tsitsiklis, J.; Bertsekas, D.; and Athans, M. 1986. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9): 803–812.

Wang, J.; and Joshi, G. 2021. Cooperative SGD: A unified framework for the design and analysis of local-update SGD algorithms. *The Journal of Machine Learning Research*, 22(1): 9709–9758.

Wang, J.; Tantia, V.; Ballas, N.; and Rabbat, M. 2019. Slowmo: Improving communication-efficient distributed sgd with slow momentum. *arXiv preprint arXiv:1910.00643*.

Wang, M.; Bodonhelyi, A.; Bozkir, E.; and Kasneci, E. 2024. TurboSVM-FL: Boosting Federated Learning through SVM Aggregation for Lazy Clients. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, 15546–15554.

Wang, S.; Pi, A.; and Zhou, X. 2019. Scalable distributed dl training: Batching communication and computation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, 5289–5296.

Wei, Y.; Zhang, Y.; Song, Q.; Zhou, X.; Zhou, Y.; and Shen, Y. 2022. Effects of different configurations and gradients on compression responses of gradient honeycombs via selective laser melting. *Thin-Walled Structures*, 170.

Wen, M.; Liu, C.; and Xu, Y. 2024. Communication Efficient Distributed Newton Method over Unreliable Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, 15832–15840.

Wu, G.; and Gong, S. 2021. Decentralised learning from independent multi-domain labels for person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, 2898–2906.

Wu, X.; Zhang, J.; and Wang, F.-Y. 2019. Stability-based generalization analysis of distributed learning algorithms for big data. *IEEE Transactions on Neural Networks and Learning Systems*, 31: 801–812.

Yan, G.; Wang, H.; and Li, J. 2022. Seizing critical learning periods in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 36, 8788–8796.

Yu, H.; Jin, R.; and Yang, S. 2019. On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In *Proceedings of International Conference on Machine Learning (ICML)*.

Yuan, K.; Chen, Y.; Huang, X.; Zhang, Y.; Pan, P.; Xu, Y.; and Yin, W. 2021. DecentLaM: Decentralized momentum SGD for large-batch deep training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Zang, Y.; Xue, Z.; Ou, S.; Chu, L.; Du, J.; and Long, Y. 2024. Efficient Asynchronous Federated Learning with Prospective Momentum Aggregation and Fine-Grained Correction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, 16642–16650.

Zhang, M.; Lucas, J.; Ba, J.; and Hinton, G. E. 2019. Lookahead optimizer: k steps forward, 1 step back. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 32.

Zhang, S.; Choromanska, A. E.; and LeCun, Y. 2015. Deep learning with elastic averaging SGD. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 28.

Zinkevich, M.; Weimer, M.; Li, L.; and Smola, A. 2010. Parallelized stochastic gradient descent. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 23.