Live Gradient Compensation for Evading Stragglers in Distributed Learning

Jian Xu*, Shao-Lun Huang*, Linqi Song†, Tian Lan‡
*Tsinghua-Berkeley Shenzhen Institute, Tsinghua University
†City University of Hong Kong, ‡George Washington University
xujian20@mails.tsinghua.edu.cn, shaolun.huang@sz.tsinghua.edu.cn, linqi.song@cityu.edu.hk, tlan@gwu.edu

***Abstract*—The training efficiency of distributed learning systems is vulnerable to stragglers, namely, those slow worker nodes. A naive strategy is performing the distributed learning by incorporating the fastest K workers and ignoring these stragglers, which may induce high deviation for non-IID data. To tackle this, we develop a Live Gradient Compensation (LGC) strategy to incorporate the one-step delayed gradients from stragglers, aiming to accelerate learning process and utilize the stragglers simultaneously. In LGC framework, mini-batch data are divided into smaller blocks and processed separately, which makes the gradient computed based on partial work accessible. In addition, we provide theoretical convergence analysis of our algorithm for non-convex optimization problem under non-IID training data to show that LGC-SGD has almost the same convergence error as full synchronous SGD. The theoretical results also allow us to quantify a novel tradeoff in minimizing training time and error by selecting the optimal straggler threshold. Finally, extensive simulation experiments of image classification on CIFAR-10 dataset are conducted, and the numerical results demonstrate the effectiveness of our proposed strategy.**

***Index Terms*—Straggler, Distributed Learning, Non-IID, Gradient Compensation**

I. Introduction

Distributed implementations of gradient-based methods [1], [2] have been essential for training large machine learning models on massive datasets, e.g., deep neural networks for image classification and speech recognition [3], [4]. Typical distributed learning architecture consists of a parameter server (PS) and distributed worker nodes – the workers compute and send local gradients to PS in parallel, while the PS aggregates the gradients and then broadcasts back to workers to update local parameters [5]. In synchronous settings, the time overhead of each iteration in such system architecture is subject to the stragglers, i.e., slow or unresponsive workers that are caused by performance variability as well as unexpected incidents like network congestion and hardware failures. It has been shown that mitigating stragglers is crucial for fully capitalizing on the benefits of distributed learning [6]–[9].

Much research attention has recently focused on mitigating stragglers either by leveraging coding-theoretic techniques [10]–[14] or by utilizing partial work completed by stragglers [15], [16]. In particular, it is possible to collect gradients from only the fast workers and discard the computations on stragglers, while still achieving convergence [8], [17], [18]. We refer to this naive strategy as K-SGD. However, this approach relies on the IID assumption[1] of training data and is shown to induce gradient/sampling bias in more general settings. Another line of work leverages gradient coding to obtain the exact gradient value despite of the stragglers [7], [9], [19]–[21]. However in such approaches, a certain amount of overhead (computation/data duplication) must always be present, in order to successfully address the worst-case stragglers. Further, gradient coding schemes are brittle in the sense that they work perfectly only up to a fixed number of stragglers.

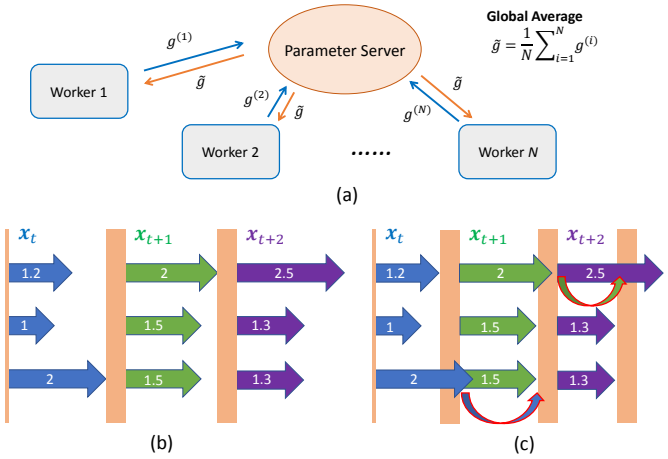


Fig. 1. Distributed learning with a parameter server. (a) System architecture. (b) Full-SGD method. (c) LGC-SGD method. LGC-SGD can significantly reduce training time overhead while maintaining convergence, by evading and compensating for stragglers.

In this paper, we propose a novel Live Gradient Compensation (LGC) framework for mitigating stragglers in distributed learning, which is built on top of K-SGD with partial work tolerance and gradient bias compensation mechanisms. The central idea can be seen through a simple example shown in Fig. 1 with one PS and three workers. In full synchronous SGD (Full-SGD), the training time overhead of each iteration is determined by the worst performing worker, which results in a total training time of $2 + 2 + 2.5 = 6.5$ for all three iterations. On the other hand, we can bypass the slowest worker in each iteration (thus collecting the results only from the $K = 2$ fastest workers) and then compensate for the impact by performing a combined gradient update in the next iteration. This reduces the total training time to $1.2 + 1.5 + 1.3 = 4$,

[1]Training data among workers are independent and identically distributed (IID), so that local gradient is an unbiased estimation of global gradient.

albeit minor gradient noise introduced due to the one-step delay of compensation. This motivates the design of LGC-SGD and its theoretical analysis. In particular, we quantify the convergence speed of LGC-SGD for arbitrary choice of K and prove that the learning algorithm is guaranteed to converge to a critical point even with non-convex objectives and non-IID training data. We would like to emphasize that in contrast to the gradient coding approach, LGC-SGD does not require any extra computation or data storage overhead. While gradient compensation has been developed as a technique in gradient compression [22]–[26], we make novel use of that to mitigate stragglers in distributed learning. We also note that asynchronous methods need extra assumption for theoretically ensuring convergence and often generates relatively high and uncertain training errors [8], [27]–[30], thus we focus on synchronous distributed learning in this paper.

To the best of our knowledge, this is the first work to use gradient compensation for mitigating stragglers on the fly. In contrast to existing approaches like directly ignoring stragglers and gradient coding, our proposed LGC framework guarantees training convergence with non-convex objectives and non-IID training data, while introducing no additional overhead for computation or data duplication. In particular, for a system with N workers and any threshold K, we quantify the gradient bias and variance induced by ignoring $N-K$ slow workers and show that their negative impact on convergence can be successfully alleviated by choosing an appropriate, one-step delayed gradient compensation that can be integrated into the next update. We show that the bias-compensated parameters enjoy a similar update rule with Full-SGD, thus the convergence analysis could be performed similarly and the same $O(1/\sqrt{NT})$ convergence rate can be achieved.

To minimize the training error, our analysis illuminates an interesting design tradeoff between efficiency and accuracy, since selecting less workers leads to more iterations under a limited training time budget while the estimated gradient value can become more accurate if more workers are allowed to finish. We quantify this tradeoff and find the "sweet spot" of optimal K (i.e., the optimal number of workers to finish in each iteration) for minimizing training error within a fixed training time budget t. This result gives us some insights to adjust K for different time budgets.

The proposed LGC framework is evaluated on CIFAR-10 dataset with various cases by changing the level of non-IID and the straggling period length, which verify the effectiveness in speeding up training while keeping a high model generalization ability. Our simulation results show that $\sim$35% saving in training time can be obtained with only slight accuracy loss. Moreover, the tradeoff by designing K is numerically characterized and compared. To summarize, the main contributions of this paper are as follows:

- A new distributed training strategy based on one-step delayed gradient compensation, namely LGC-SGD, is proposed for evading stragglers and utilizing partial work.
- We quantify the convergence property of LGC-SGD by characterizing the gradient bias and through order statistic analysis, which illuminates a new design tradeoff.
- Theoretical convergence analysis of proposed algorithm for non-convex optimization on non-IID data is provided.
- The effectiveness of proposed LGC-SGD is verified on CIFAR-10 dataset, where LGC-SGD can significantly reduce training time while converging to the same training error compared with Full-SGD.

II. Related Work

In [8], [17], the K-SGD is directly employed to mitigate the impact of stragglers. In contrast to fixed K, distributed SGD with adaptive K is also investigated in [8], [18], in which the value of K gradually increased throughout the training process. However, they do not consider persistent stragglers nor deal with non-IID data. In [8], [21], it's pointed out that if training data is IID among workers, ignoring stragglers is less harmful, otherwise it would have a negative impact on convergence. As to distributed learning on non-IID data, related work include the federated learning [23], [31], [32]. In [32], the well-known FedAvg method is theoretically proved to work for federated learning on non-IID data, but the assumption of unbiased device sampling and parameter averaging scheme are essential for obtaining the results. In this paper we consider both IID and non-IID cases in distributed learning with stragglers through a novel LGC-SGD method.

In [15], a scheme called Anytime Minibatch is designed to exploit the partial work completed by slow worker nodes. In this manner local gradient is averaged over the actually computed stochastic gradients within the constrained computation time. However, the fixed computation time is empirical and varies with task and mini-batch size, and the non-convex and non-IID cases were not studied. Partial work combining coding is also explored in [16] through multi-message communication strategy, where each worker is assigned with multiple data partitions and would send a couple of recently calculated gradients for multiple times in a communication round. Though it can utilize the stragglers but data redundancy is still needed, notably increasing the learning overhead. Our solution is partly motivated by these work but a live compensation strategy with provable convergence without adding computation/storage redundancy.

III. The Proposed LGC Framework

A. System Model for Distributed Learning

We focus on distributed optimization of a non-convex problem on non-IID data. We assume that training data are distributed over multiple worker nodes in a network, and all workers jointly optimize a shared model based on local data. Mathematically, the underlying distributed optimization problem can be formalized as follows:

$$\min_{\mathbf{x}\in R^d} F(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\xi_i\sim D_i}\left[F(\mathbf{x};\xi_i)\right] \tag{1}$$

where N is the number of workers, D_i denotes the local dataset of i-th worker and could have different distribution

from other workers (which means the IID assumption is relaxed), and $F(\mathbf{x};\xi_i)$ denotes the local loss function given shared model parameters $\mathbf{x}$ and training data ξ_i (one sample point or a mini-batch) sampled from D_i of the i-th worker.

We make all workers initialized to the same point $\mathbf{x_0}$, then Full-SGD can be employed to solve the problem. At each iteration, the i-th worker randomly draws a mini-batch samples ξ_i from D_i, and computes local stochastic gradient with respect to global shared parameter $\mathbf{x}_t$:

$$g_t^{(i)} = g(\mathbf{x}_t;\xi_i) = \frac{1}{|\xi_i|}\sum_{j=1}^{|\xi_i|}\nabla F(\mathbf{x}_t;\xi_i^{(j)}) \tag{2}$$

The parameter server aggregates all the local gradients to get a global gradient:

$$\tilde{g}_t = \frac{1}{N}\sum_{i=1}^{N} g_t^{(i)} \tag{3}$$

Then the result will be broadcast to all worker nodes to update their local models and start a new iteration. This process will repeat until the model converges.

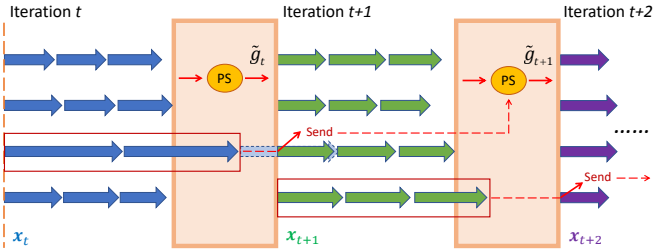


Fig. 2. Illustration of the workflow of proposed LGC-SGD. Mini-batch data are divided into multiple blocks and processed in order.

B. Our Proposed Solution

The proposed LGC framework is described in Algorithm 1 and the training process is illustrated in Fig. 2. Specifically, for each worker, mini-batch of data are divided into s smaller blocks and computed incrementally. Slow worker may not be able to completely finish its task by next iteration, but perhaps it has processed r of s blocks and can send an approximate result afterwards. At each iteration, the server collects the fastest K fresh gradients that evaluated on entire mini-batch as well as any delayed gradients from the previous iteration, and obtains a parameter update by combining average fresh gradient and a proper compensation for gradient bias of the previous iteration. The gradient bias induced by ignoring stragglers is quantified by Eq. (7). Meanwhile, the remaining slow workers are allowed to continue computing until the entire mini-batch are evaluated or new global update is received, after which the delayed gradients are sent to the server for bias compensation. It is worth noting that the delayed gradients could be computed based on full mini-batch data or a portion, depending on the computation speed of stragglers. Suppose the i-th worker computed r blocks of samples within an iteration

Algorithm 1 Live Gradient Compensation SGD

Input: learning rate η, total iteration T, partition number s, mini-batch size m, total workers N, threshold K
Initial: $\mathbf{x}_0 \in R^d$; $e_{-1} = 0$
for $t = 0, 1, ..., T-1$ **do**
  On each worker *i* :
  divide mini-batch samples ξ_i into s partitions
  $g_t^{(i)} = 0, r_t^{(i)} = 1$
  while $r_t^{(i)} \leq s$ **and** update not received **do**
    $g_t^{(i)} = g_t^{(i)} + g(\mathbf{x}_t;\xi_i[r_t^{(i)}])$
    $r_t^{(i)} = r_t^{(i)} + 1$
  end while
  send $g_t^{(i)} = g_t^{(i)}/s$ to server
  wait for global update $\tilde{g}_t$ from server
  update local model: $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\tilde{g}_t$
  On server:
  collect fastest K gradients from workers
  average: $\tilde{g_t}' = \frac{1}{K}\sum_{i\in S_t} g_t^{(i)}$
  if $t \geq 1$ **then**
    collect delayed $(N-K)$ gradients from stragglers
    calculate compensation:
    $e_{t-1} = \frac{1}{N}\sum_{i\notin S_{t-1}} g_{t-1}^{(i)} - \frac{N-K}{N}\tilde{g}'_{t-1}$
  end if
  obtain the global update: $\tilde{g}_t = \tilde{g_t}' + e_{t-1}$
  send $\tilde{g}_t$ to all workers
end for

and the mini-batch size is m, the variance of local stochastic gradient evaluated on a sample point is bounded by σ^2. Then the expectation and variance of gradient satisfy the following[2]:

$$\mathbb{E}\left[g_t^{(i)}\right] = \nabla F_i(\mathbf{x}_t) \tag{4}$$

$$\mathbb{E}\left[\left\|g_t^{(i)} - \nabla F_i(\mathbf{x}_t)\right\|^2\right] \leq \frac{s}{r}\cdot\frac{\sigma^2}{m} \tag{5}$$

It means that the gradient based on partial work is still a reliable estimation, equivalent to that obtained by scaling down mini-batch size. Therefore, if gradients of stragglers could be measured and compensated, the training performance can be guaranteed. That is the main motivation of the proposed strategy. Considering the enlarged variance may influence the training process, we adopt a linear scaling rule on the gradient to address this issue as Eq. (6), which is similar to the linear scaling rule on the learning rate as [33]. The scaling operation may reduce the magnitude of gradient, but not affect the estimation of direction, having the effect of variance reduction.

$$g_t^{(i)} = \frac{r}{s}\cdot\frac{1}{r}\sum_{k=1}^{r} g(\mathbf{x}_t;\xi_i[k]) \tag{6}$$

[2]In this paper, $\|\cdot\|$ denotes the ℓ_2 norm.

Let $\tilde{g}_t$ and $\tilde{g}'_t$ denote the average gradient of N and K workers respectively, and S_t the set of fastest K workers. Then the gradient bias caused by ignoring stragglers can be obtained as follows:

$$e_t = \tilde{g}_t - \tilde{g}'_t = \frac{1}{N}\sum_{i=1}^{N} g_t^{(i)} - \frac{1}{K}\sum_{k\in S_t} g_t^{(k)}$$
$$= \frac{N-K}{N}\left[\frac{1}{N-K}\sum_{k\notin S_t} g_t^{(k)} - \frac{1}{K}\sum_{k\in S_t} g_t^{(k)}\right] \quad (7)$$

The first term in parentheses in the last step yields the average gradient of stragglers, and the second term is the aforementioned average gradient of fastest K workers.

IV. Theoretical Analysis of LGC

In this section, we provide the theoretical analysis of the proposed LGC-SGD for non-convex optimization problem, jointly considering the non-IID training data and persistent straggling behavior. We investigate the convergence properties of both K-SGD and LGC-SGD by iteratively analyzing the sequence of gradient update. Besides, we quantify a novel tradeoff in minimizing training error under given training time budget by selecting different straggler threshold K. We collect all theorem proofs in the Appendix.

A. Preliminaries

We first make the following basic assumption, which is commonly used in the literature [25], [34], [35] for convergence analysis of distributed optimization.

Assumption 1. *Assume that problem (1) satisfies:*

1. ***Smoothness****: The objective function $F(\cdot)$ is smooth with Lipschitz constant $L > 0$, which means $\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\| \leq L\|\mathbf{x}-\mathbf{y}\|, \forall \mathbf{x}, \forall \mathbf{y}$. It implies that:*

$$F(\mathbf{x}) - F(\mathbf{y}) \leq \nabla F(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \frac{L}{2}\|\mathbf{x}-\mathbf{y}\|^2 \quad (8)$$

2. ***Unbiased local gradient****: For each worker with local data, the stochastic gradient is locally unbiased:*

$$\mathbb{E}_{\xi_i\sim D_i}[\nabla F(\mathbf{x};\xi_i)] = \nabla F_i(\mathbf{x}) \quad (9)$$

3. ***Bounded variances****: The stochastic gradient evaluated on a sample point of each worker has a bounded variance uniformly, satisfying:*

$$\mathbb{E}_{\xi_i\sim D_i}[\|\nabla F(\mathbf{x};\xi_i) - \nabla F_i(\mathbf{x})\|^2] \leq \sigma^2 \quad (10)$$

and the deviation between local and global gradient satisfies:

$$\|\nabla F_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 \leq \beta^2 \quad (11)$$

The above assumptions are valid for both IID and non-IID cases. We do not assume that the worker nodes can access the same dataset and thus consider the non-IID problem in particular. The β^2 in third assumption quantifies the deviation between local and global gradient, and we consider that $\beta^2 = 0$ for IID case and $\beta^2 > 0$ for non-IID case.

B. Convergence Analysis

We first provide the following useful lemma to characterize the gradient bias caused by initially ignoring computations of stragglers, which can help analyze convergence properties of both K-SGD and LGC-SGD under non-IID training data. We note that the analysis of K-SGD provides the basis as well as valuable insights for analyzing LGC-SGD.

Lemma 1. *Suppose the training data among workers are non-IID under Assumption 1, then the deviation between the average gradient of all workers and the average gradient of any K workers can be characterized as follows:*

$$\mathbb{E}[\|\tilde{g} - \tilde{g}'\|^2] \leq \delta^2 + \frac{N-K}{NK}\sigma^2 \quad (12)$$

and if the global gradient is estimated based on the results on K workers, the variance can be bounded as:

$$\mathbb{E}\left[\|\tilde{g}' - \nabla F(\mathbf{x})\|^2\right] \leq \delta^2 + \frac{1}{K}\sigma^2 \quad (13)$$

where $\delta^2 = \min\{\beta^2, \frac{(N-K)^2}{K^2}\beta^2\}$ depends on the value of K.

Proof. Detailed proof of Lemma 1 is in Appendix A. □

Remark 1. *The extra term δ^2 in Lemma 1 arises from non-IID, which is the main reason for obtaining a different gradient variance than in IID case. It is worth noting that when $K = N$ we always have $\delta^2 = 0$, and that is the reason why Full-SGD still works well for non-IID problem. We recommend choosing $K > 0.5N$ for non-IID problem to reduce the δ^2. When the problem is optimized by mini-batch SGD with batch-size m, it is not hard to show that σ^2 can be substituted by σ^2/m but the δ^2 remains unchanged.*

The notations in Lemmas 1 will be directly used later in this paper to perform the convergence analysis. For basic K-SGD, if we assume that the stragglers are random and independent across iterations, extending the analysis technique for IID case in [8], [35] we can characterize the convergence of K-SGD by Theorem 1.

Theorem 1. *For problem (1) under Assumption 1, suppose that the fastest K workers are random and independent at each iteration, and that the K-SGD method employs a fixed learning rate $\eta \leq 1/L$ and $F^* = \min_{\mathbf{x}} F(\mathbf{x})$, then we have the following convergence result:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \leq \frac{2(F(\mathbf{x}_0) - F^*)}{\eta T} + L\eta(\delta^2 + \frac{\sigma^2}{K}) \quad (14)$$

Proof. Detailed proof of Theorem 1 is in Appendix B. □

The above theorem shows that given enough training iteration T and small learning rate η, the algorithm converges to a critical point. When the training data among workers are IID, the result is straightforward and holds even when the fastest K workers are not random due to persistent stragglers. But for non-IID training data, the assumption of random selection

of K workers is essentially needed. However, in practice it is possible for some worker nodes to remain as stragglers for an extended period of time. Thus, we relax this assumption to further investigate the convergence under persistent stragglers and non-IID training data. Based on Lemma 1, we can revise the error bound in Theorem 1 and get the following result.

Proposition 1. *For problem (1) under Assumption 1, suppose that the K-SGD method employs a fixed learning rate $\eta \leq 1/4L$ and $F^* = \min_{\mathbf{x}} F(\mathbf{x})$. Then we have the following convergence result:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \leq \frac{2(F(\mathbf{x}_0)-F^*)}{\eta T} + 2L\eta(\delta^2+\frac{\sigma^2}{K}) + 2(\delta^2+\frac{N-K}{NK}\sigma^2) \quad (15)$$

Proof. Detailed proof of Proposition 1 is in Appendix C. □

Comparing the results of Theorem 1 and Proposition 1, it can be seen that *K*-SGD may result in higher training error after convergence when stragglers are persistent. Because there exists extra constant term on the right hand side of Proposition 1, which does not diminish even under small learning rate. To address this issue, we utilize the delayed gradients of stragglers to generate compensation in the next round of global update. We provide the following convergence theorem for LGC-SGD.

Theorem 2. *For problem (1) under Assumption 1 and let $F^* = \min_{\mathbf{x}} F(\mathbf{x})$, if LGC-SGD emplys a fixed learning rate $\eta \leq 1/2L$, then we have the following convergence result:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \leq \frac{2(F(\mathbf{x}_0)-F^*)}{\eta T} + \frac{L\eta\sigma^2}{N} + 2L^2\eta^2(\delta^2+\frac{N-K}{NK}\sigma^2) \quad (16)$$

Proof. Detailed proof of Theorem 2 is in Appendix D. □

Based on Theorem 2, We can obtain the $O(1/\sqrt{NT})$ convergence rate by appropriately choosing the learning rate, as illustrated in the following corollary. When T is sufficiently large, $O(1/\sqrt{NT})$ dominates $O(N/T)$, so the acceleration of convergence rate is almost free from the influence of K.

Corollary 1. *For problem (1) under Assumption 1 and let $F^* = \min_{\mathbf{x}} F(\mathbf{x})$, if LGC-SGD employs a fixed learning rate $\eta = \frac{\sqrt{N}}{\sqrt{T}}$, then for any $T \geq 4NL^2$, we have the following convergence result:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] = O\left(\frac{1}{\sqrt{NT}}\right) + O\left(\frac{N}{T}\right) \quad (17)$$

Theorem 2 shows that the last term in the upper bound in Eq. (16) is proportional to the squared learning rate, which makes it different from K-SGD result in Proposition 1. It means that deviation in the learned model parameters due to gradient bias can be made arbitrarily small when choosing a sufficiently small learning rate, enabling LGC-SGD to achieve the same convergence error as Full-SGD. At this point, the above convergence analysis theoretically demonstrates that the improved algorithm can overcome the drawbacks of naive K-SGD by taking advantage of live gradient compensation completed by stragglers.

C. Tradeoff by Selecting Threshold K

The selection of threshold *K* illuminates a new trade-off in training time and convergence error, which is crucial in distributed learning. Intuitively, under the same conditions, smaller values of *K* make each iteration complete more quickly but result in higher gradient bias while bigger values of *K* lead to longer per-iteration time but lower gradient bias. Thus there exists a "sweet spot" of the choice of K to achieve the smallest training error under fixed training time budget. In the previous analysis, we have obtained the training error for given number of iterations T. As to training time analysis, we need to find the per-iteration time distribution through order statistics [8], [18], [36]. Suppose that the wall-clock time to complete each iteration can be described by random variable X_i for worker i, where X_i's are i.i.d. across iterations and workers. Then, the expected time spent at each iteration for the fastest K workers is the expectation of the K^{th} order statistic of N i.i.d. random variables $X_1, X_2, ..., X_N$, denoted by $\mathbb{E}\left[X_{(K)}\right]$. We consider the shifted exponential distribution, which have been widely used in the literature [6], [8], [16] to model the per-iteration computation time. Let $\bar{\mu}$ and μ_K denote the expectation of single variable and the K^{th} order statistic respectively, and let τ denote the averaged communication latency, t the total training time constraint. Substituting the iteration number T with $t/(\tau+\mu_K)$ and choosing $K > 0.5N$, then $\delta^2 \leq \frac{(N-K)^2}{K^2}\beta^2 \leq \frac{N-K}{K}\beta^2$ and the bound in Theorem 2 can be rewritten as the following function of *K*:

$$B(K) = \frac{2(F(\mathbf{x}_0)-F^*)}{\eta t}\mu_K + \frac{2L^2\eta^2\left(N\beta^2+\sigma^2\right)}{K} + const \quad (18)$$

Before finding the optimal K to minimize training error, we propose two basic constraints on the selection of K as follows:

$$\mu_K \leq (1+\lambda)\bar{\mu}, \quad 0<\lambda<1 \quad (19)$$

$$\frac{\mu_N}{s} \leq \mu_K + \tau, \quad 1 < s \leq m \quad (20)$$

The first constraint ensures the per-iteration computation time would not exceed the average execution time of single worker by $\lambda\mu$. The second constraint means that stragglers should have computed at least one data block before parameters are updated. This is needed because otherwise the gradient from slowest worker would be unattainable.

Take shifted exponential distribution $X_i \sim \Delta + Exp(\mu)$ and further mathematical analysis on the threshold selection strategy can be carried out. For large value of N, we have the following approximations of expected order statistic [8]:

$$\mathbb{E}[X_{(K)}] = \Delta + \mu\log\frac{N}{N-K} \quad (K<N) \quad (21)$$

$$\mathbb{E}[X_{(N)}] = \Delta + \mu\log N \quad (22)$$

Since the communication latency and shift Δ are not known in advance, here we obtain the feasible lower- and upper-bounds of appropriate K:

$$\max\left\{0.5, 1-e^{-\left(\frac{\log N}{s}\right)}\right\} \leq \frac{K}{N} \leq 1-e^{-(1+\lambda)} \tag{23}$$

Empirical results in [20], [37] show that in large cluster of workers, only around 2% nodes require much longer computation time than the median. Therefor, the upper bound of K controlled by small λ is sufficient to evade stragglers. To minimize the $B(K)$ with respect to K, take its derivative and we have:

$$\frac{dB(K)}{dK} = \frac{2\mu(F(\mathbf{x}_0)-F^*)}{\eta t(N-K)} - \frac{2L^2\eta^2(\beta^2+\sigma^2/N)N}{K^2} \tag{24}$$

Setting the derivative to 0, we get a quadratic function :

$$\frac{1-p}{p^2} = \frac{\mu(F(\mathbf{x}_0)-F^*)}{L^2\eta^3 t(\beta^2+\sigma^2/N)} \tag{25}$$

where $p = \frac{K}{N}$ is constrained as in (23). It's obvious that the second order derivative of $B(K)$ is positive, so the optimal selection of K is unique.

While the exact value of optimal K cannot be readily computed using Eq. (25), it depends on the optimal value of objective, Lipschitz constant and gradient variance that are not known in advance, we can still get some important insights from Eq. (25). It is not hard to show that the optimal p will increase with larger the time budget t and for IID problems with $\beta^2 = 0$, and will decrease with growing system size N. The result also reveals that the optimal value of p increases as the objective function gets closer to the optimal value. If dividing the total training process into multiple phases, then the best strategy would be to gradually increase the value of p (and thus K) over time. This makes sense since in the initial phase small K helps to reduce the gap between initial value and optimal value of objective function quickly, and in the later phase big K should be used to aim for smaller convergence error. This analysis result provides a theoretical support for the heuristic algorithms previously developed in [8], [18]. It can be leveraged to construct more advanced strategies for tuning K and N with respect to training time/error objectives.

V. Simulation

The proposed LGC-SGD approach and theoretical results are evaluated on the CIFAR-10 dataset [38]. We implemented the learning task in a non-IID distributed manner. Since data duplication and coded computation may be unfeasible in some cases, such as federated learning, we only evaluate the Full-SGD and naive K-SGD as baselines for comparison.

A. Experimental Setup

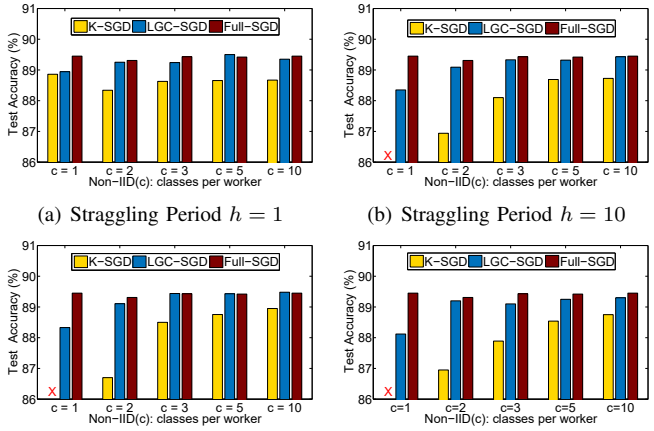
Dataset and Model. The well-known CIFAR-10 dataset contains 10 object classes with 50,000 training samples and 10,000 testing samples. Here we use the notation non-IID(c) to mean that each worker is allocated with c categories of samples. We constructed our model based on VGG-11 [39], where we adjusted the neural network to fit the input size and kept only one fully connected layer without dropout layer.

Simulation Setting. To simulate the straggling behaviors, we use shifted exponential distribution to generate the per-iteration computation time, on which the stragglers are identified. The mini-batch size is set to 32 and each training algorithm is run for total 60 epochs. The initial learning rate is set to 0.1 and divided by 10 after 30 epochs. The momentum is set to 0.9 and weight decay is set to 0.0005. For K-SGD and LGC-SGD, the first 2 epochs are run in Full-SGD fashion as warmup. All algorithms are implemented in PyTorch.

Metrics. We trained the model for a fixed number of epochs and use the training loss curve as well as model test accuracy to evaluate the training performance. Specifically, loss value after training and time consumption are utilized as metrics to evaluate training efficiency. Average test accuracy of last 5 epochs is used to assess the model generalization ability.

B. Numerical Results

We conduct simulations for N = 10 workers and choose K = 7 as the straggler threshold value. As [21] we introduce dependency between stragglers across iterations by fixing per-iteration computation time for h iterations, after which the computation time for each worker will be generated randomly and independently again. To begin with, we simply assume that the stragglers can complete all computations before the beginning of next iteration. We repeated each experiment for three times and reported the average result. Fig. 3 shows the main results of model test accuracy under different level of non-IID for different straggling behaviors.

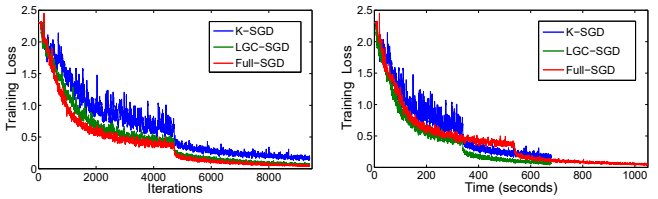


(a) Straggling Period $h = 1$ (b) Straggling Period $h = 10$

(c) Straggling Period $h = 50$ (d) Straggling Period $h = 150$

Fig. 3. Test accuracy comparison of training methods under various level of non-IID and different straggling period length. LGC-SGD outperforms K-SGD and catches up with Full-SGD.

1) Robustness to Non-IID: We reduce the value of c to generate data distributions with increasing non-IID level and test the robustness of LGC-SGD. It can be found that Full-SGD performs well despite of data skewness among workers. However, the K-SGD has lower test accuracy, especially under large data skewness and persistent straggling behavior, such as non-IID(2) and non-IID(3) in Fig. 3(b)(c)(d), and even diverges under non-IID(1). In contrast, the proposed



(a) Training loss v.s. iterations (b) Training loss v.s. time

Fig. 4. The convergence of training loss over iterations and time. LGC-SGD has faster training speed and comparable convergence error than Full-SGD.

LGC-SGD can effectively leverage gradient compensation to eliminate gradient bias during the training process, achieving comparable model generalization ability as Full-SGD after the same number of training iterations.

2) Robustness to Straggling Period: We also change the h to simulate different straggling behavior to investigate the impact on training. Fig. 3 provides the test results of 3 cases, where $h = 1$ means the stragglers are random and independent every iteration while $h = 10$ means that stragglers are randomly selected every 10 iterations. It's interesting to see that when $h = 1$, the test result of K-SGD is less affected by non-IID, verifying our result in Theorem 1. As we increase the value of h, the model trained by K-SGD results in lower test accuracy due to gradient bias induced by discarding the computation of stragglers. However, the model trained by LGC-SGD still achieves nearly equal test result to Full-SGD.

3) Efficiency Improvement: Take the case of $c = 3$ and $h = 10$, the convergence of training loss in terms of the number of iterations and generated wall-clock time are plotted in Fig. 4, where we use $X_i \sim 0.05 + Exp(0.02)$ to generate and simulate per-iteration time. The Full-SGD can achieve lowest convergence error at the cost of longer overall training time, while K-SGD can save per-iteration time but result in higher convergence error. However, the LGC-SGD can have the best of both worlds by significantly reduce training time as K-SGD while achieving almost the same training loss as Full-SGD. The simulation result demonstrates that LGC-SGD can reduce training time by up to $\sim$35% compared with the Full-SGD, while achieving nearly the same convergence error.

C. Discussions

Finally, we perform analysis on other factors that may affect the performance of LGC-SGD. Specifically, we evaluate the behaviors of LGC-SGD with different choices of K, different percentages of the mini-batch data that are processed by stragglers for each iteration as well as different system sizes. In this part, we fix $c = 3$ and $h = 10$ while the results are similar for other values of c and h.

1) Tradeoff through K: As mentioned previously, the selection of threshold K is non-trivial and highlights an important tradeoff between minimizing training error and training time. We gradually increased the value of K from 5 to 10 for fixed number of iterations to plot the optimal frontier of training time and training loss as Fig. 5, in which different colors represent different values of K and the red digital labels denote

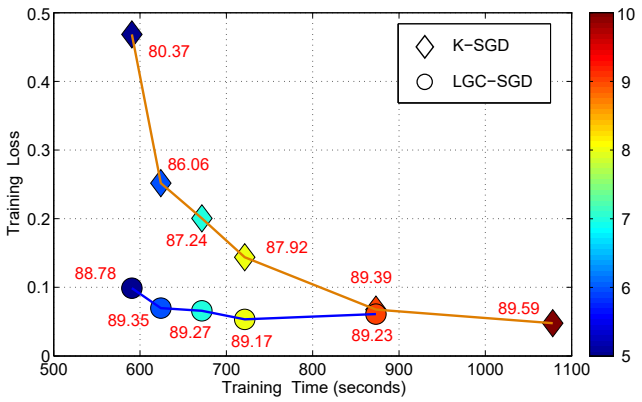


Fig. 5. Training loss and training time as well as test accuracy for various value of K under fixed training iterations.

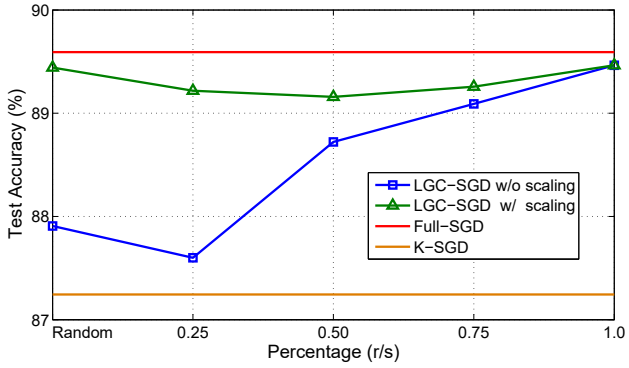


Fig. 6. Tolerating partial work by linear scaling on the gradient of stragglers.

test accuracy. It can be found that as K decreases, the model test accuracy of K-SGD degrades substantially while LGC-SGD only has slight accuracy loss. It experimentally reveals that the selection of K is an explicit tradeoff bewteen training time and model accuracy. And the optimal frontier achieved by LGC-SGD significantly improves that of K-SGD.

2) Partial Work: We artificially make the slow workers only process different percentages of mini-batch data to study the influence of tolerating partial work of stragglers. We keep the batch-size as $m = 32$ and set the number of blocks as $s = 4$, then simulations are conducted under fixed and random amount of partial work ($r = 1 \sim 4$). Particularly, we compare the results of LGC-SGD with and without linear scaling on the delayed gradients of stragglers as shown in Fig. 6. It can be seen that LGC-SGD with linear scaling on delayed gradient can effectively utilize the partial work of stragglers.

3) Scalability: To further verify the effectiveness, we compare the results of three different system sizes, where we set $N = 10, 20, 40$ with $K = 0.7N$ respectively and adjust the batch-size to keep $mN = 320$. It is observed that LGC-SGD consistently achieves nearly optimal test accuracy (i.e., 89.27, 89.22, 89.28 vs. 89.59, 89.49, 89.43 in Full-SGD for $N = 10, 20, 40$) and significant saving in training time over Full-SGD (i.e., $\sim$35%, $\sim$40%, $\sim$45% for $N = 10, 20, 40$), which indicate the resilience of LGC-SGD.

VI. Conclusion

In this work, we proposed a live gradient compensation framework to evade stragglers in distributed learning system. It can overcome the drawbacks of naively ignoring stragglers in synchronous SGD and unlike gradient coding approaches does not require any extra computation/storage overhead. We particularly investigated the performance of LGC-SGD on non-IID training data, providing theoretical analysis on the convergence error and quantifying the tradeoff by selecting different straggler threshold value. Simulation results on CIFAR-10 dataset verified our theoretical findings and demonstrated the effectiveness of proposed LGC-SGD. Future work includes developing strategies to dynamically adjust different hyper-parameters in LGC-SGC in practical distributed systems.

Acknowledgment

Prof. Shao-Lun Huang is supported by the National Natural Science Foundation of China (61807021), Shenzhen Science and Technology Program (KQTD20170810150821146), Innovation and Entrepreneurship Project for Overseas High-Level Talents of Shenzhen (KQJSCX20180327144037831). Prof. Linqi Song is supported by the Hong Kong RGC grant ECS 21212419, and Guangdong Basic and Applied Basic Research Foundation under Key Project 2019B1515120032.

References

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[2] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng, "Large scale distributed deep networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1223–1231.

[3] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[5] M. Li, D. G. Andersen, A. Smola, and K. Yu, "Communication efficient distributed machine learning with the parameter server," *Advances in Neural Information Processing Systems (NIPS)*, vol. 1, pp. 19–27, 2014.

[6] J. Dean and L. A. Barroso, "The tail at scale," *Communications of The ACM*, vol. 56, no. 2, pp. 74–80, 2013.

[7] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *International Conference on Machine Learning (ICML)*, 2017.

[8] S. Dutta, G. Joshi, S. Ghosh, P. Dube, and P. Nagpurkar, "Slow and stale gradients can win the race: Error-runtime trade-offs in distributed sgd," in *AISTATS*, 2018.

[9] C. Karakus, Y. Sun, S. Diggavi, and W. Yin, "Redundancy techniques for straggler mitigation in distributed optimization and learning," *Journal of Machine Learning Research*, vol. 20, no. 72, pp. 1–47, 2019.

[10] C. Karakus, Y. Sun, S. N. Diggavi, and W. Yin, "Straggler mitigation in distributed optimization through data encoding," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 5434–5442.

[11] S. Li, S. M. M. Kalan, A. S. Avestimehr, and M. Soltanolkotabi, "Near-optimal straggler mitigation for distributed gradient methods," in *IEEE International Parallel and Distributed Processing Symposium Workshops(IPDPS)*, 2018, pp. 857–866.

[12] K. Lee, M. Lam, R. Pedarsani, D. S. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, 2018.

[13] H. Park, K. W. Lee, J.-Y. Sohn, C. Suh, and J. Moon, "Hierarchical coding for distributed computing," in *IEEE International Symposium on Information Theory (ISIT)*, 2018, pp. 1630–1634.

[14] S. Li, S. M. M. Kalan, Q. Yu, M. Soltanolkotabi, and A. S. Avestimehr, "Polynomially coded regression: Optimal straggler mitigation via data encoding," *arXiv:1805.09934*, 2018.

[15] N. Ferdinand, H. Al-Lawati, S. Draper, and M. Nokleby, "Anytime minibatch: Exploiting stragglers in online distributed optimization," in *International Conference on Learning Representations (ICLR)*, 2019.

[16] E. Ozfatura, D. Gündüz, and S. Ulukus, "Speeding up distributed gradient descent by utilizing non-persistent stragglers," in *IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 2729–2733.

[17] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous sgd," *arXiv:1604.00981*, 2016.

[18] S. Kas Hanna, R. Bitar, P. Parag, V. Dasari, and S. El Rouayheb, "Adaptive distributed stochastic gradient descent for minimizing delay in the presence of stragglers," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[19] M. Ye and E. Abbe, "Communication-computation efficient gradient coding," in *International Conference on Machine Learning (ICML)*, 2018, pp. 5606–5615.

[20] H. Wang, Z. B. Charles, and D. S. Papailiopoulos, "Erasurehead: Distributed gradient descent without delays using approximate gradient coding," *arXiv:1901.09671*, 2019.

[21] R. Bitar, M. Wootters, and S. El Rouayheb, "Stochastic gradient coding for straggler mitigation in distributed learning," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 277–291, 2020.

[22] S. U. Stich, J. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[23] F. Sattler, S. Wiedemann, K. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2019.

[24] S. Zheng, Z. Huang, and J. T. Kwok, "Communication-efficient distributed blockwise momentum sgd with error-feedback," in *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[25] H. Tang, C. Yu, X. Lian, T. Zhang, and J. Liu, "`DoubleSqueeze`: Parallel stochastic gradient descent with double-pass error-compensated compression," in *International Conference on Machine Learning (ICML)*, 2019, pp. 6155–6165.

[26] S. U. Stich and S. P. Karimireddy, "The error-feedback framework: Sgd with delayed gradients," *Journal of Machine Learning Research*, vol. 21, no. 237, pp. 1–36, 2020.

[27] F. Niu, B. Recht, C. Re, and S. J. Wright, "Hogwild! a lock-free approach to parallelizing stochastic gradient descent," in *Advances in Neural Information Processing Systems (NIPS)*, 2011, pp. 693–701.

[28] S. Zhang, C. Zhang, Z. You, R. Zheng, and B. Xu, "Asynchronous stochastic gradient descent for dnn training," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[29] X. Lian, Y. Huang, Y. Li, and J. Liu, "Asynchronous parallel stochastic gradient for nonconvex optimization," in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 2737–2745.

[30] H. R. Feyzmahdavian, A. Aytekin, and M. Johansson, "An asynchronous mini-batch algorithm for regularized stochastic optimization," *IEEE Trans. Automat. Contr.*, vol. 61, no. 12, pp. 3740–3754, 2016.

[31] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv:1806.00582*, 2018.

[32] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *International Conference on Learning Representations (ICLR)*, 2020.

[33] P. Goyal, P. Dollár, R. B. Girshick, and P. Noordhuis, "Accurate, large minibatch SGD: training imagenet in 1 hour," *arXiv:1706.02677*, 2017.

[34] H. Yu, R. Jin, and S. Yang, "On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization," in *International Conference on Machine Learning (ICML)*, vol. 97, 2019, pp. 7184–7193.

[35] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, 2018.

[36] J. Wang and G. Joshi, "Adaptive communication strategies to achieve the best error-runtime trade-off in local-update SGD," in *Proceedings of Machine Learning and Systems (MLSys)*, 2019.

[37] V. Gupta, D. Carrano, Y. Yang, V. Shankar, T. A. Courtade, and K. Ramchandran, "Serverless straggler mitigation using local error-correcting codes," *arXiv:2001.07490*, 2020.

[38] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *University of Toronto*, 2009.

[39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

APPENDIX

A. Proof of Lemma 1

Given a arbitrary subset of workers S_t, which represents the fastest K workers, then

$$\begin{aligned}
&\mathbb{E}[\left\|\tilde{g}_t - \tilde{g}_t{}'\right\|^2] \\
&= \mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N} g_t^{(i)} - \frac{1}{K}\sum_{j\in S_t} g_t^{(j)}\right\|^2\right] \\
&= \mathbb{E}\left[\left\|\frac{N-K}{N}\frac{1}{N-K}\sum_{i\notin S_t}^{N} g_t^{(i)} - \frac{N-K}{N}\frac{1}{K}\sum_{j\in S_t} g_t^{(j)}\right\|^2\right]
\end{aligned}$$

Let $\mathbf{A} = \sum_{i\notin S_t}\left(g_t^{(i)} - \nabla F(\mathbf{x}_t)\right)$, $\mathbf{B} = \sum_{j\in S_t}\left(g_t^{(j)} - \nabla F(\mathbf{x}_t)\right)$, then $\mathbf{A}$ and $\mathbf{B}$ are independent. We have $\mathbb{E}[\mathbf{A}+\mathbf{B}] = \mathbf{0}$ and

$$\mathbb{E}[\left\|\tilde{g}_t - \tilde{g}_t{}'\right\|^2] = \frac{(N-K)^2}{N^2}\mathbb{E}\left[\left\|\frac{1}{N-K}\mathbf{A} - \frac{1}{K}\mathbf{B}\right\|^2\right]$$

Recall that σ^2 is the bounded local variance for local gradient and β^2 is bounded deviation between local and global gradient. Applying the Jensen inequality, we have

$$\left\|\mathbb{E}\left[\mathbf{A}\right]\right\|^2 \le (N-K)\sum_{i\notin S_t}\left\|\nabla F_i(\mathbf{x}_t) - \nabla F(\mathbf{x}_t)\right\|^2 \le (N-K)^2\beta^2$$

$$\left\|\mathbb{E}\left[\mathbf{B}\right]\right\|^2 \le K\sum_{i\in S_t}\left\|\nabla F_i(\mathbf{x}_t) - \nabla F(\mathbf{x}_t)\right\|^2 \le K^2\beta^2$$

Notice that $\mathbb{E}[\mathbf{A}] = -\mathbb{E}[\mathbf{B}]$, thus

$$\left\|\mathbb{E}\left[\mathbf{A}\right]\right\|^2 = \left\|\mathbb{E}\left[\mathbf{B}\right]\right\|^2 \le \min\{(N-K)^2\beta^2, K^2\beta^2\}$$

Using the basic relation between expectation and variance, we have

$$\begin{aligned}
\mathbb{E}\left\|\mathbf{A}\right\|^2 &= \left\|\mathbb{E}[\mathbf{A}]\right\|^2 + \mathrm{var}[\mathbf{A}] \le \left\|\mathbb{E}[\mathbf{A}]\right\|^2 + (N-K)\sigma^2 \\
\mathbb{E}\left\|\mathbf{B}\right\|^2 &= \left\|\mathbb{E}[\mathbf{B}]\right\|^2 + \mathrm{var}[\mathbf{B}] \le \left\|\mathbb{E}[\mathbf{B}]\right\|^2 + K\sigma^2
\end{aligned}$$

Based on the above relations, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\frac{1}{N-K}\mathbf{A} - \frac{1}{K}\mathbf{B}\right\|^2\right] \\
&= \mathbb{E}\left\|\frac{1}{N-K}\mathbf{A}\right\|^2 + \mathbb{E}\left\|\frac{1}{K}\mathbf{B}\right\|^2 - 2\mathbb{E}\left\langle\frac{1}{N-K}\mathbf{A}, \frac{1}{K}\mathbf{B}\right\rangle \\
&\le \left[\frac{1}{N-K} + \frac{1}{K}\right]^2\left\|\mathbb{E}\left[\mathbf{A}\right]\right\|^2 + \left[\frac{1}{N-K} + \frac{1}{K}\right]\sigma^2 \\
&\le \frac{N^2}{(N-K)^2K^2}\left\|\mathbb{E}\left[\mathbf{A}\right]\right\|^2 + \frac{N}{(N-K)K}\sigma^2
\end{aligned}$$

Combining the above results together, we get

$$\begin{aligned}
\mathbb{E}[\left\|\tilde{g}_t - \tilde{g}_t{}'\right\|^2] &= \frac{1}{K^2}\left\|\mathbb{E}\left[\mathbf{A}\right]\right\|^2 + \frac{N-K}{NK}\sigma^2 \\
&\le \begin{cases}\beta^2 + \frac{N-K}{NK}\sigma^2, & K \le 0.5N \\ \frac{(N-K)^2}{K^2}\beta^2 + \frac{N-K}{NK}\sigma^2, & K > 0.5N\end{cases}
\end{aligned}$$

Since

$$\mathbb{E}\left\|\mathbf{B}\right\|^2 = \min\{(N-K)^2\beta^2, K^2\beta^2\} + K\sigma^2$$

we directly have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\tilde{g}_t{}' - \nabla F(\mathbf{x}_t)\right\|^2\right] = \frac{1}{K^2}\mathbb{E}\left\|\mathbf{B}\right\|^2 \\
&\le \min\{\beta^2, \frac{(N-K)^2}{K^2}\beta^2\} + \frac{1}{K}\sigma^2 \\
&\le \begin{cases}\beta^2 + \frac{1}{K}\sigma^2, & K \le 0.5N \\ \frac{(N-K)^2}{K^2}\beta^2 + \frac{1}{K}\sigma^2, & K > 0.5N\end{cases}
\end{aligned}$$

It completes the proof of Lemma 1.

B. Proof of Theorem 1

The averaged gradient of fastest K workers is unbiased estimation of global gradient under Assumption 1, whether for IID or non-IID training data. Taking the total expectations of averaged gradient on local sampling and workers selection, we have

$$\begin{aligned}
\mathbb{E}\left[\tilde{g}_t{}'\right] &= \mathbb{E}\left[\frac{1}{K}\sum_{i\in S_t} g_t^{(i)}\right] = \frac{1}{\binom{N}{K}}\sum_{j=1}^{\binom{N}{K}}\mathbb{E}\left[\frac{1}{K}\sum_{i\in S_j} g_t^{(i)}\right] \\
&= \frac{1}{K\cdot\binom{N}{K}}\sum_{i=1}^{N}\binom{N-1}{K-1}\mathbb{E}\left[g_t^{(i)}\right] \\
&= \frac{N\cdot\binom{N-1}{K-1}}{K\cdot\binom{N}{K}}\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[g_t^{(i)}\right] \\
&= \nabla F(\mathbf{x}_t)
\end{aligned}$$

Then, by the smoothness of objective function F, we have

$$\begin{aligned}
&\mathbb{E}_t[F(\mathbf{x}_{t+1})] - F(\mathbf{x}_t) \\
&\le \nabla F(\mathbf{x}_t)^T\mathbb{E}_t[\mathbf{x}_{t+1} - \mathbf{x}_t] + \frac{L}{2}\mathbb{E}_t\left[\left\|\mathbf{x}_{t+1} - \mathbf{x}_t\right\|^2\right] \\
&= -\eta\left\langle\nabla F(\mathbf{x}_t), \mathbb{E}_t\left[\tilde{g}_t'\right]\right\rangle + \frac{L\eta^2}{2}\mathbb{E}_t\left[\left\|\tilde{g}_t'\right\|^2\right] \\
&= -\eta\left\|\nabla F(\mathbf{x}_t)\right\|^2 + \frac{L\eta^2}{2}\left\|\nabla F(\mathbf{x}_t)\right\|^2 \\
&\qquad + \frac{L\eta^2}{2}\mathbb{E}_t\left[\left\|\frac{1}{K}\sum_{i\in S_t} g_t^{(i)} - \nabla F(\mathbf{x}_t)\right\|^2\right] \\
&= -\eta(1 - \frac{L\eta}{2})\left\|\nabla F(\mathbf{x}_t)\right\|^2 + \frac{L\eta^2}{2}(\delta^2 + \frac{\sigma^2}{K})
\end{aligned}$$

Taking total expectation and rearrange the terms, we have

$$\begin{aligned}
&\eta(1 - \frac{L\eta}{2})\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \\
&\qquad \le \mathbb{E}[F(\mathbf{x}_t) - F(\mathbf{x}_{t+1})] + \frac{L\eta^2}{2}(\delta^2 + \frac{\sigma^2}{K})
\end{aligned}$$

Assume that $\eta \leq \frac{1}{L}$, thus $1 - \frac{L\eta}{2} \geq \frac{1}{2}$. Taking summation and dividing by $\eta(1 - \frac{L\eta}{2})T$, then we get

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \\
&\leq \frac{\sum_{t=0}^{T-1}\mathbb{E}[F(\mathbf{x}_t) - F(\mathbf{x}_{t+1})]}{\eta(1-\frac{L\eta}{2})T} + \frac{L\eta}{2(1-\frac{L\eta}{2})}(\delta^2 + \frac{\sigma^2}{K}) \\
&\leq \frac{2(F(\mathbf{x}_0) - F^*)}{\eta T} + L\eta(\delta^2 + \frac{\sigma^2}{K}),
\end{aligned}$$

which completes the proof.

C. Proof of Proposition 1

Similar to the proof of Theorem 1, we have

$$\begin{aligned}
&\mathbb{E}_t[F(\mathbf{x}_{t+1})] - F(\mathbf{x}_t) \\
&\leq -\eta\left\langle \nabla F(\mathbf{x}_t), \mathbb{E}_t\left[\tilde{g}_t'\right]\right\rangle + \frac{L\eta^2}{2}\mathbb{E}_t\left[\|\tilde{g}_t'\|^2\right] \\
&= -\eta\left\langle \nabla F(\mathbf{x}_t), \mathbb{E}_t\left[\tilde{g}_t\right]\right\rangle + \eta\left\langle \nabla F(\mathbf{x}_t), \mathbb{E}_t\left[\tilde{g}_t - \tilde{g}_t'\right]\right\rangle \\
&\quad + \frac{L\eta^2}{2}\mathbb{E}_t\left[\|\tilde{g}_t' - \nabla F(\mathbf{x}_t) + \nabla F(\mathbf{x}_t)\|^2\right] \\
&\leq -\eta\|\nabla F(\mathbf{x}_t)\|^2 + \frac{\eta\rho}{2}\|\nabla F(\mathbf{x}_t)\|^2 + \frac{\eta}{2\rho}\mathbb{E}[\|\tilde{g}_t - \tilde{g}_t'\|^2] \\
&\quad + L\eta^2\|\nabla F(\mathbf{x}_t)\|^2 + L\eta^2\mathbb{E}_t\left[\|\tilde{g}_t' - \nabla F(\mathbf{x}_t)\|^2\right] \\
&\leq -\frac{\eta}{2}(2 - \rho - 2L\eta)\|\nabla F(\mathbf{x}_t)\|^2 \\
&\quad + \frac{\eta}{2\rho}(\delta^2 + \frac{N-K}{NK}\sigma^2) + L\eta^2(\delta^2 + \frac{\sigma^2}{K})
\end{aligned}$$

where the Young's inequality with $\rho > 0$, the basic inequality $(\mathbf{a}+\mathbf{b})^2 \leq 2(\mathbf{a}^2 + \mathbf{b}^2)$ and Lemma 1 are applied. Choosing $\rho = 0.5$, taking total expectation and rearranging the terms, we get

$$\begin{aligned}
\eta(\frac{3-4L\eta}{4})\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] &\leq \mathbb{E}[F(\mathbf{x}_t) - F(\mathbf{x}_{t+1})] \\
&\quad + L\eta^2(\delta^2 + \frac{\sigma^2}{K}) + \eta(\delta^2 + \frac{N-K}{NK}\sigma^2)
\end{aligned}$$

Assume that $\eta \leq 1/4L$, thus $\frac{3-4L\eta}{4} \geq \frac{1}{2}$. Taking summation and dividing by $\eta\frac{(3-4L\eta)}{4}T$, then we get

$$\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] &\leq \frac{4\sum_{t=0}^{T-1}\mathbb{E}[F(\mathbf{x}_t) - F(\mathbf{x}_{t+1})]}{\eta(3-4L\eta)T} \\
&\quad + \frac{4(L\eta(\delta^2 + \frac{\sigma^2}{K}) + (\delta^2 + \frac{N-K}{NK}\sigma^2))}{3-4L\eta} \\
&\leq \frac{2(F(\mathbf{x}_0) - F^*)}{\eta T} + 2L\eta(\delta^2 + \frac{\sigma^2}{K}) + 2(\delta^2 + \frac{N-K}{NK}\sigma^2)
\end{aligned}$$

which completes the proof.

D. Proof of Theorem 2

We first show that a modified parameter sequence $\hat{\mathbf{x}}_t = \mathbf{x_t} - \eta e_{t-1}$ satisfies $\hat{\mathbf{x}}_{t+1} = \hat{\mathbf{x}}_t - \eta\tilde{g}_t$, where $e_{t-1} = \tilde{g}_{t-1} - \tilde{g}_{t-1}'$. According to the algorithm, we have

$$\begin{aligned}
\mathbf{x}_{t+1} &= \mathbf{x}_t - \eta(\tilde{g}_t' + e_{t-1}) \\
&= \mathbf{x}_t - \eta(\tilde{g}_t - (\tilde{g}_t - \tilde{g}_t')) - \eta e_{t-1} \\
&= \mathbf{x}_t - \eta e_{t-1} - \eta_t\tilde{g}_t + \eta e_t
\end{aligned}$$

therefore, we get

$$(\mathbf{x}_{t+1} - \eta e_t) = (\mathbf{x}_t - \eta e_{t-1}) - \eta\tilde{g}_t$$

Based on above result and by the smoothness, we have

$$\begin{aligned}
&\mathbb{E}_t[F(\hat{\mathbf{x}}_{t+1})] - F(\hat{\mathbf{x}}_t) \\
&\leq \nabla F(\hat{\mathbf{x}}_t)^T\mathbb{E}_t[\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t] + \frac{L}{2}\mathbb{E}_t\left[\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\|^2\right] \\
&= -\eta\left\langle \nabla F(\hat{\mathbf{x}}_t), \mathbb{E}_t\left[\tilde{g}_t\right]\right\rangle + \frac{L\eta^2}{2}\mathbb{E}_t\left[\|\tilde{g}_t\|^2\right] \\
&\leq -\eta\left\langle \nabla F(\hat{\mathbf{x}}_t), \nabla F(\mathbf{x}_t)\right\rangle + \frac{L\eta^2}{2}\left[\|\nabla F(\mathbf{x}_t)\|^2 + \frac{\sigma^2}{N}\right]
\end{aligned}$$

Applying Young's inequality with $\rho > 0$ and by the smoothness, we have

$$\begin{aligned}
&-\eta\left\langle \nabla F(\hat{\mathbf{x}}_t), \nabla F(\mathbf{x}_t)\right\rangle \\
&= -\eta\left\langle \nabla F(\mathbf{x}_t), \nabla F(\mathbf{x}_t)\right\rangle + \eta\left\langle \nabla F(\mathbf{x}_t) - \nabla F(\hat{\mathbf{x}}_t), \nabla F(\mathbf{x}_t)\right\rangle \\
&\leq -\eta(1 - \frac{\rho}{2})\|\nabla F(\mathbf{x}_t)\|^2 + \frac{\eta}{2\rho}\|\nabla F(\mathbf{x}_t) - \nabla F(\hat{\mathbf{x}}_t)\|^2 \\
&\leq -\eta(1 - \frac{\rho}{2})\|\nabla F(\mathbf{x}_t)\|^2 + \frac{L^2\eta}{2\rho}\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2 \\
&= -\eta(1 - \frac{\rho}{2})\|\nabla F(\mathbf{x}_t)\|^2 + \frac{L^2\eta^3}{2\rho}\|e_{t-1}\|^2
\end{aligned}$$

Choosing $\rho = 0.5$, taking total expectation and rearranging the terms, we get

$$\begin{aligned}
&\eta(\frac{3-2L\eta}{4})\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \\
&\leq \mathbb{E}[F(\hat{\mathbf{x}}_t) - F(\hat{\mathbf{x}}_{t+1})] + \frac{L\eta^2\sigma^2}{2N} + L^2\eta^3(\delta^2 + \frac{N-K}{NK}\sigma^2)
\end{aligned}$$

Choosing a fixed learning rate $\eta \leq 1/2L$, thus $\frac{3-2L\eta}{4} \geq \frac{1}{2}$. Taking summation and dividing by $\eta\frac{(3-2L\eta)}{4}T$, then we get

$$\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] &\leq \frac{4\sum_{t=0}^{T-1}\mathbb{E}[F(\hat{\mathbf{x}}_t) - F(\hat{\mathbf{x}}_{t+1})]}{\eta(3-2L\eta)T} \\
&\quad + \frac{4(\frac{L\eta\sigma^2}{2N} + L^2\eta^2(\delta^2 + \frac{N-K}{NK}\sigma^2))}{3-2L\eta} \\
&\leq \frac{2(F(\mathbf{x}_0) - F^*)}{\eta T} + \frac{L\eta\sigma^2}{N} + 2L^2\eta^2(\delta^2 + \frac{N-K}{NK}\sigma^2)
\end{aligned}$$

which completes the proof.