# Lower Bounds and Nearly Optimal Algorithms in Distributed Learning with Communication Compression

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Recent advances in distributed optimization and learning have shown that communication compression is one of the most effective means of reducing communication. While there have been many results for convergence rates with compressed communication, a lower bound is still missing.

Analyses of algorithms with communication compression have identified two abstract properties that guarantee convergence: the unbiased property or the contractive property. They can be applied either unidirectionally (compressing messages from worker to server) or bidirectionally. In the smooth and non-convex stochastic regime, this paper establishes a lower bound for distributed algorithms whether using unbiased or contractive compressors in unidirection or bidirection. To close the gap between this lower bound and the best existing upper bound, we further propose an algorithm, NEOLITHIC, that almost reaches our lower bound (except for a logarithm factor) under mild conditions. Our results also show that using contractive compressors in bidirection can yield iterative methods that converge as fast as those using unbiased compressors unidirectionally. We report experimental results that validate our findings.

## 1 Introduction

Large-scale optimization is a critical step in many machine learning applications. Millions or even billions of data samples contribute to the excellent performance in tasks such as robotics, computer vision, natural language processing, healthcare, and so on. However, such a scale of data samples and model parameters leads to enormous communication that hampers the scalability of distributed machine-learning training systems. We urgently need communication-reduction strategies. State-of-the-art strategies include model and gradient compression [4, 13, 51], decentralized communication [39, 10, 21], lazy communication [67, 52, 33, 20], and beyond. This article will focus on the former.

The most common method of distributed training is Parallel SGD (P-SGD) [16]. In P-SGD, the stochastic gradients that workers transmit to a server cause significant communication overhead in large-scale machine learning. To reduce this overhead, many recent works propose to compress the messages sent unidirectionally from worker to server [5, 29, 4] or compress the messages between them bidirectionally [57, 64]. The method of compression is either sparsification or quantization [5, 29, 4] or their combination [29, 15]. The literature [53, 50, 57, 64] reveals that bidirectional compression can save more communication, but leads to slower convergence rates.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Table 1: Comparison between various distributed stochastic algorithms with communication compression for non-convex loss functions. To explicitly clarify the influence of different compression strategies, we keep the stochastic gradient variance $\sigma^2$, data heterogeneity bound $b^2$ , gradient square norm bound $G^2$ (used in [53, 57, 62], much larger than $b^2$ and $L$) , but omit smoothness constant $L$, and initialization $f(x^{(0)}) - f^\star$ in the below results. Moreover, notation $\tilde{\mathcal{O}}(\cdot)$ hides all logarithmic terms. "LB" and "UB" indicate lower and upper bound, respectively. Notation $\delta$ and $\omega$ are compressor-related parameters (see detailed discussions in Sec. 2).

| | **Algorithm** | **Convergence Rate** | **Compression**[a] | **Trans. Compl.**[b] |
|---|---|---|---|---|
| LB | Theorem 1+Corollary 1 | $\Omega\left(\frac{\sigma}{\sqrt{nT}}+\frac{1}{\delta T}\right)$ | Uni/Bidirectional Contractive | $\mathcal{O}(n/\delta^2)$ |
| | Theorem 2+Corollary 2 | $\Omega\left(\frac{\sigma}{\sqrt{nT}}+\frac{1+\omega}{T}\right)$ | Uni/Bidirectional Unbiased | $\mathcal{O}\left(n(1+\omega)^2\right)$ |
| UB | Theorem 3 | $\tilde{\mathcal{O}}\left(\frac{\sigma}{\sqrt{nT}}+\frac{1}{\delta T}\right)$ | Uni/Bidirectional Contractive | $\tilde{\mathcal{O}}(n/\delta^2)$ |
| | Corollary 3 | $\tilde{\mathcal{O}}\left(\frac{\sigma}{\sqrt{nT}}+\frac{1+\omega}{T}\right)$ | Uni/Bidirectional Unbiased | $\tilde{\mathcal{O}}(n(1+\omega)^2)$ |
| | Q-SGD [32] | $\mathcal{O}\left(\frac{(1+\omega)\sigma+\omega b^2/\sigma}{\sqrt{nT}}\right)^\dagger$ | Unidirectional i.i.d, Unbiased | – |
| | MEM-SGD [53] | $\mathcal{O}\left(\frac{\sigma}{\sqrt{nT}}+\frac{G^{2/3}}{\delta^{2/3}T^{2/3}}+\frac{1}{T}\right)$ | Unidirectional Contractive | $\mathcal{O}(n^3/\delta^4)$ |
| | Double Squeeze [57] | $\mathcal{O}\left(\frac{1}{\sqrt{nT}}+\frac{G^{2/3}}{\delta^{4/3}T^{2/3}}+\frac{1}{T}\right)$ | Bidirectional$^\diamond$ | $\mathcal{O}(n^3/\delta^8)$ |
| | CSER [62] | $\mathcal{O}\left(\frac{\sigma}{\sqrt{nT}}+\frac{G^{2/3}}{\delta^{2/3}T^{2/3}}+\frac{1}{T}\right)$ | Unidirectional Contractive | $\mathcal{O}(n^3/\delta^4)$ |
| | EF21-SGD [25] | $\mathcal{O}\left(\frac{\sigma}{\sqrt{\delta^3 T}}+\frac{1}{\delta T}\right)^*$ | Unidirectional Contractive | – |

[a] This column indicates the type of the compressor and in what direction the compression is applied.

[b] This column indicates the transient complexity, *i.e.*, the number of gradient queries (or communication rounds) the algorithm has to experience before reaching the linear-speedup stage, *i.e.*, $\mathcal{O}(\frac{\sigma}{\sqrt{nT}})$.

[†] This convergence rate is valid only for $T=\Omega(n(1+\frac{\omega}{n})^2)$. Since the rate $\frac{(1+\omega)\sigma+\omega b^2/\sigma}{\sqrt{nT}}$ is always worse than $\frac{\sigma}{\sqrt{nT}}$, the transient complexity is not available.

[*] Since the convergence rate does not show linear-speedup, the transient complexity are not available

Although there are many specific compression methods, their convergence analyses are mainly built on unbiased compressibility or contractive compressibility. The literature [15, 49, 65] summarizes these two properties and how they appear in the analyses. An unbiased compressor compresses a long vector $x$ into a short vector $C(x)$ and satisfies $\mathbb{E}[C(x)] = x$, i.e., no bias is introduced. The contractive compressor may introduce bias, but its compression introduces much less variance. We give their definitions below. Although the contractive compressor can empirically work better, the analysis of the unbiased compressor yields faster convergence due to unbiasedness [15, 42, 30, 27].

Despite the quick progress made in compression techniques and their convergence, we do not yet understand the limits of algorithms with communication compression. Since unbiased and contractive compressibilities are the two representative characteristics of various compressors, we use them to theorize two types of compression methods. For each type, we intend to answer: *What is the optimal convergence rate that a distributed algorithm can achieve when using any of the compression methods of this type?* Here, we assume that only unbiased compressibility or contractive compressibility can be used, not considering any additional special compressor design; after all, any special design in the literature has been heuristic, whose effectiveness can be explained at best and not proved or quantified. So, we further clarify our question: *Given a class of optimization problems (specified below) and a class of compression methods of the same type, if we choose the worst combination of them to defeat an algorithm, what will the convergence rate that the best-defending algorithm can reach?* To our knowledge, they are fundamental questions not addressed yet.

**1.1 Main Results** This paper clarifies these open questions by providing lower bounds under the non-convex smooth stochastic optimization setting, and developing effective algorithms that match the lower bounds up to logarithm factors. In particular, our contributions are:

- We establish convergence lower bounds for distributed algorithms with communication compression in the stochastic non-convex regime. Our lower bounds apply to any algorithm conducting unidirectional or bidirectional compression and using unbiased or contractive compressors. We find a clear gap between the established lower bounds and the existing convergence rates.
- We propose a novel **ne**arly **o**ptimal a**l**gor**ith**m w**i**th **c**ompression (NEOLITHIC) to fill in this gap. NEOLITHIC can adopt either unidirectional or bidirectional compression, and is compatible with both unbiased and contractive compressors. Using any combination, NEOLITHIC provably matches the above lower bound, under an additional mild assumption and up to logarithmic factors.
- The convergence results of NEOLITHIC also imply that algorithms using biased contractive compressors bidirectionally can theoretically converge as fast as those with unbiased compressors used unidirectionally. Before our work, it is established in [15, 53, 50, 57, 64] that algorithms with unidirectional compression and unbiased compressors enjoy better convergence rates.
- We provide extensive experimental results to validate our theories.

All established results in this paper as well as convergence rates of existing state-of-the-art distributed algorithms with communication compression are listed in Table 1. The transient complexity, which measures how sensitive the algorithm is to the compression strategy (see Sec. 3), is also listed in the table. The smaller the transient complexity is, the faster the algorithm converges.

**1.2 Related Works.** Due to the page limit, we can only describe the most closely related works.

**Distributed learning.** Distributed learning has been increasingly useful in training large-scale machine learning models [22]. It typically follows a centralized or decentralized setup. Centralized approaches [2, 38], with P-SGD as the representative, require all workers to synchronize with a central server per iteration. Decentralized approaches, however, are based on partial averaging in which each worker only needs to synchronize with its immediate neighbors. Well-known decentralized algorithms include decentralized SGD [43, 19, 69, 39, 10], $D^2$[56, 68], and stochastic gradient tracking [63, 36, 3]. The lazy communication [67, 52, 20] is also utilized to reduce communication overheads in which workers conducts multiple local updates before sending messages.

**Communication compression.** To alleviate the communication overhead in distributed learning, researchers have proposed two mainstream communication compression methods: quantization and sparsification. Quantization [32, 55, 72] is essentially an unbiased operator with random noise. For example, [51] develops Sign-SGD by using only 1 bit for each entry whose convergence is studied in [13, 14, 60]. Q-SGD [4] compresses each entry with more flexible bits and enables a trade-off between convergence rates and communication costs. Sparsification, on the other hand, amounts to a biased but contractive operator. [59, 53] propose to transmit a small number of entries, randomly or by taking the largest ones, to achieve sparsity in communication. The theories behind contractive compressors are limited to those in [58, 40, 54] due to analysis challenges, and they are established with assumptions such as bounded gradients [73, 34] or quadratic loss functions [61]. More discussions on unbiased and biased compressors can be found in [15, 49].

**Error compensation.** The error compensation (feedback) mechanism is introduced by [51] to mitigate the error caused by 1-bit quantization. [61] studies SGD with error-compensated quantization for quadratic functions with convergence guarantees. [53] shows that error compensation can reduce quantization-incurred errors for strongly convex loss functions in the single-node setting. Error-compensated SGD is studied in [5] for non-convex loss functions with no establishment of an improved convergence rate. The algorithm EF21 [48] applies to the deterministic setting, using contractive compressors unidirectionally. It can converge under very mild assumptions. It has been recently extended to the stochastic setting [25]. We don't have a linear-speedup result for EF21 (yet).

**Lower bounds in optimization.** Lower bounds are well studied in convex optimization [24, 1, 23] especially when there is no gradient noise [9, 70, 11, 6, 26]. In non-convex optimization, [17, 18] propose a zero-chain model and show a tight bound for first-order methods. By splitting the zero-chain model into multiple components, the authors of [74, 8] extended the approach to finite sum and stochastic problems. Recently, [41] shows a lower bound in the decentralized setting (for the linear

graph) based on the approach. However, no lower bound result exists for distributed learning with communication compression to our knowledge.

## 2 Problem Setup

We consider standard distributed learning with $n$ parallel workers. The data samples in each worker $i$ follow a local distribution $D_i$, which can be heterogeneous among all workers. These workers, together with a central parameter server, collaborate to train a model by solving

$$\min_{x\in\mathbb{R}^d} \quad f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) \quad \text{with} \quad f_i(x) = \mathbb{E}_{\xi_i\sim D_i}[F(x;\xi_i)], \tag{1}$$

where $F(x;\xi)$ is the loss function evaluated at parameter $x$ with sample $\xi$. Since the objective $f$ can be non-convex, finding a global minimum of (1) is generally intractable. Therefore, we turn to seeking a model $\hat{x}$ with a small gradient magnitude in expectation, *i.e.*, $\mathbb{E}[\|\nabla f(\hat{x})\|^2]$. Next we introduce the setup under which we perform a convergence analysis.

**2.1 Function Class.** We let the function class $\mathcal{F}_{\Delta,L}$ denote the set of all functions satisfying Assumption 1 for any underlying dimension $d \in \mathbb{N}_+$ and a given initialization point $x^{(0)} \in \mathbb{R}^d$.

**Assumption 1 (Smoothness).** *We assume each local objective $f_i$ has $L$-Lipschitz gradient,* i.e.*,*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \le L\|x-y\|$$

*for any $i \in \{1,\dots,n\}, x, y \in \mathbb{R}^d$, and $f(x^{(0)}) - \inf_{x\in\mathbb{R}^d} f(x) \le \Delta$ with $f = \frac{1}{n}\sum_{i=1}^n f_i$.*

**2.2 Gradient Oracle Class.** We assume each worker $i$ has access to its local gradient $\nabla f_i(x)$ via a stochastic gradient oracle $O_i(x;\zeta_i)$ subject to randomness $\zeta_i$, *e.g.*, the mini-batch sampling $\zeta_i \triangleq \xi_i \sim D_i$. We further assume that the output $O_i(x,\zeta_i)$ is a time-independent and unbiased estimator of the full-batch gradient $\nabla f_i(x)$ with a bounded variance. Formally, we let the stochastic gradient oracle class $\mathcal{O}_{\sigma^2}$ denote the set of all oracles $O_i$ satisfying Assumption 2.

**Assumption 2 (Gradient noise).** *For any $x \in \mathbb{R}^d$ and $i \in \{1,\dots,n\}$, the oracle $O_i$ satisfies*

$$\mathbb{E}_{\zeta_i}[O_i(x;\zeta_i)] = \nabla f_i(x) \quad \textit{and} \quad \mathbb{E}_{\zeta_i}[\|O_i(x;\zeta_i) - \nabla f_i(x)\|^2] \le \sigma^2.$$

**2.3 Compressor Class.** The two classes of widely-used compressors are i) the $\omega$-*unbiased* compressor, described by Assumption 3, *e.g.*, the stochastic quantization operator [4], and ii) the $\delta$-*contractive* compressor, described by Assumption 4, *e.g.*, the rand-$K$ and top-$K$ operators [53, 47].

**Assumption 3 (Unbiased compressor).** *For a (possibly random) compression operator $C : \mathbb{R}^d \to \mathbb{R}^d$, we assume there exists a constant $\omega \ge 0$ such that*

$$\mathbb{E}[C(x)] = x, \quad \mathbb{E}[\|C(x)-x\|^2] \le \omega\|x\|^2, \quad \forall\, x \in \mathbb{R}^d,$$

*where the expectation is taken over the randomness of the compression operator $C$.*

**Assumption 4 (Contractive compressor).** *For a (possibly random) compression operator $C : \mathbb{R}^d \to \mathbb{R}^d$, we assume there exists a constant $\delta \in (0,1]$ such that*

$$\mathbb{E}[\|C(x)-x\|^2] \le (1-\delta)\|x\|^2, \quad \forall\, x \in \mathbb{R}^d,$$

*where the expectation is over the randomness of the compression operator $C$.*

Formally, we let the compressor classes $\mathcal{U}_\omega$ and $\mathcal{C}_\delta$ denote the set of all $\omega$-unbiased compressors and $\delta$-contractive compressors satisfying Assumptions 3 and 4, respectively. Note that the identity operator $I$ satisfies $I \in \mathcal{U}_\omega$ for any $\omega \ge 0$ and $I \in \mathcal{C}_\delta$ for any $\delta \in (0,1]$. Generally, an $\omega$-unbiased compressor is not necessarily contractive when $\omega$ is larger than 1. However, since $C \in \mathcal{U}_\omega$ implies $(1+\omega)^{-1}C \in \mathcal{C}_{(1+\omega)^{-1}}$, the scaled unbiased compressor is also contractive though the converse does not hold. Hence, the class of contractive compressors is strictly more general since it contains all unbiased compressors through scaling.

**2.4 Algorithm Class.** We consider a centralized and synchronous algorithm $A$ in which i) workers are allowed to communicate only directly with the central server but not between one another; ii)

all iterations are synchronized so that all workers start each of their iterations simultaneously. Each worker $i$ holds a local copy of the model, denoted by $x_i^{(t)}$, at iteration $t$. The output $\hat{x}^{(t)}$ of $A$ after $t$ iterations can be any linear combination of all previous local models, namely,

$$\hat{x}^{(t)} \in \text{span}\left(\{x_i^{(s)} : 0 \le s \le t,\, 1 \le i \le n\}\right).$$

We further require algorithms $A$ to satisfy the so-called "zero-respecting" property, which appears in [17, 18, 41]. Intuitively, this property implies that the number of non-zero entries of the local model of a worker can be increased only by conducting local stochastic gradient descent with its own samples or synchronizing with the server. The zero-respecting property holds in all algorithms in Table 1 and most first-order methods based on SGD [44, 35, 31, 71]. In addition to these properties, the algorithm $A$ has to admit communication compression. Specifically, we endow the server with a compressor $C_0$ and each worker $i \in \{1, \cdots, n\}$ with a compressor $C_i$. If $C_i = I$ for some $i \in \{0, \cdots, n\}$, then worker $i$ (or the server if $i = 0$) conducts lossless communication. When $C_i \neq I$ for any $i \in \{0, \cdots, n\}$, algorithm $A$ conducts bidirectional compression. When $C_0 = I$, algorithm $A$ conducts unidirectional compression on messages from workers to server. The formal definition of the algorithm class with bidirectional/unidirectional compression is as follows.

**Definition 1 (Algorithm class).** *Given compressors $\{C_0, C_1, \ldots, C_n\}$, write $\mathcal{A}^B_{\{C_i\}_{i=0}^n}$ for the set of all centralized, synchronous, zero-respecting algorithms admitting bidirectional compression in which i) compressor $C_i$, $\forall\, 1 \le i \le n$, is applied to messages from worker $i$ to the server, and ii) compressor $C_0$ is applied to messages from the server to all workers. When $C_0 = I$, we write $\mathcal{A}^B_{\{C_0=I\}\cup\{C_i\}_{i=1}^n}$, or $\mathcal{A}^U_{\{C_i\}_{i=1}^n}$ for short, for the set of algorithms admitting unidirectional compression. The superscript $B$ or $U$ indicates "bidirectional" or "unidirectional", respectively.*

# 3 Lower Bounds

With all interested classes introduced above, we now define the lower bound measure. Given local loss functions $\{f_i\}_{i=1}^n \subseteq \mathcal{F}_{\Delta,L}$, stochastic gradient oracles $\{O_i\}_{i=1}^n \subseteq \mathcal{O}_{\sigma^2}$ (with $O_i$ for worker $i$), compressors $\{\mathcal{C}_i\}_{i=0}^n \subseteq \mathcal{C}$ ($\mathcal{C}$ can be $\mathcal{U}_\omega$ or $\mathcal{C}_\delta$), and an algorithm $A \in \mathcal{A}$ to solve problem (1) ($\mathcal{A}$ can be $\mathcal{A}^B_{\{C_i\}_{i=0}^n}$ or $\mathcal{A}^U_{\{C_i\}_{i=1}^n}$), we let $\hat{x}_{A,\{f_i\}_{i=1}^n,\{O_i\}_{i=1}^n,\{C_i\}_{i=0}^n,T}$ denote the output of algorithm $A$ using no more than $T$ gradient queries and rounds of communication by each worker node. We define the minimax measure

$$\inf_{A\in\mathcal{A}} \sup_{\{C_i\}_{i=0}^n\subseteq\mathcal{C}} \sup_{\{O_i\}_{i=1}^n\subseteq\mathcal{O}_{\sigma^2}} \sup_{\{f_i\}_{i=1}^n\subseteq\mathcal{F}_{\Delta,L}} \mathbb{E}[\|\nabla f(\hat{x}_{A,\{f_i\}_{i=1}^n,\{O_i\}_{i=1}^n,\{C_i\}_{i=0}^n,T})\|^2]. \tag{2}$$

In (2), we do not require the compressors $\{C_i\}_{i=0}^n$ to be distinct or independent. When $\mathcal{C}$ is $\mathcal{U}_\omega$ or $\mathcal{C}_\delta$, we allow the compressor parameter $\omega$ or $\delta$ to be accessible by algorithm $A$.

**3.1 Unidirectional Unbiased Compressors.** Our first result is for algorithms that admit unidirectional compression and $\omega$-unbiased compressors.

**Theorem 1 (Unidirectional unbiased compression).** *For every $\Delta,\, L > 0$, $n \ge 2$, $\omega \ge 0$, $\sigma > 0$, $T \ge (1+\omega)^2$, there exists a set of local loss functions $\{f_i\}_{i=1}^n \subseteq \mathcal{F}_{\Delta,L}$, stochastic gradient oracles $\{O_i\}_{i=1}^n \subseteq \mathcal{O}_{\sigma^2}$, $\omega$-unbiased compressors $\{C_i\}_{i=0}^n \subseteq \mathcal{U}_\omega$ with $C_0 = I$, such that for any algorithm $A \in \mathcal{A}^U_{\{C_i\}_{i=1}^n}$ starting from a given constant $x^{(0)}$, it holds that*

$$\mathbb{E}[\|\nabla f(\hat{x}_{A,\{f_i\}_{i=1}^n,\{O_i\}_{i=1}^n,\{C_i\}_{i=0}^n,T})\|^2] = \Omega\left(\left(\frac{\Delta L\sigma^2}{nT}\right)^{\frac{1}{2}} + \frac{(1+\omega)\Delta L}{T}\right). \tag{3}$$

**Consistency with previous works.** The bound in (3) is consistent with best-known lower bounds in different settings. When $\omega = 0$, our result reduces to the tight bounds for distributed training without compression [8]. When $n = 1$ and $\omega = 0$, our result reduces to the lower bound established in [7] under the single-node non-convex stochastic setting. When $n = 1, \omega = 0$ and $\sigma^2 = 0$, our result recovers the tight bound for deterministic non-convex optimization [17].

**Linear-speedup.** When $T$ is sufficiently large, the first term $1/\sqrt{nT}$ dominates the lower bound (3). If an algorithm achieves an $\mathcal{O}(1/\sqrt{nT})$ rate, it will require $T = \mathcal{O}(1/(n\epsilon^2))$ gradient queries to

reach a desired accuracy $\epsilon$, which is inversely proportional to $n$. Therefore, an algorithm achieves linear-speedup at $T$th iteration if, for this $T$, the term involving $nT$ is dominating the rate.

**Transient complexity.** Due to the compression-incurred overhead in convergence rate, a distributed stochastic algorithm with communication compression has to experience a transient stage to achieve its linear-speedup stage. Transient complexity are referred to the number of gradient queries (or communication rounds) when $T$ is relatively small so non-$nT$ terms still dominate the rate. The smaller the transient complexity is, the less gradient queries or communication rounds the algorithm requires to achieve the linear-speedup stage. For example, if an algorithm can achieve the lower bound established in (3), it requires $(\frac{\Delta L\sigma^2}{nT})^{\frac{1}{2}} \geq \frac{(1+\omega)\Delta L}{T}$, i.e., $T = \mathcal{O}(n(1+\omega)^2)$ transient gradient queries (or communication rounds) to achieve linear-speedup, which is proportional to the compression-related terms $(1+\omega)^2$. Transient complexity is an important metric to evaluate how sensitive the algorithm is to compression errors. It was widely used in decentralized learning [46, 66] to gauge how the network topology can influence the convergence rate.

**3.2 Bidirectional Unbiased Compressors.** Theorem 1 applies to unidirectional compression where $C_0 = I$. We next consider the bidirectional compression with $C_0 \in \mathcal{U}_\omega$. Since $\{C_i\}_{i=1}^n \subseteq \mathcal{U}_\omega$ with $C_0 = I$ is a special case of $\{C_i\}_{i=0}^n \subseteq \mathcal{U}_\omega$, the lower bound for algorithms that admit bidirectional compression is greater than or equal to that with unidirectional compression due to definition in (2).

**Corollary 1 (Bidirectional unbiased compression).** *Under the same setting as in Theorem 1, there exists a set of local objectives $\{f_i\}_{i=1}^n \subseteq \mathcal{F}_{\Delta,L}$, stochastic gradient oracles $\{O_i\}_{i=1}^n \subseteq \mathcal{O}_{\sigma^2}$, $\omega$-unbiased compressors $\{C_i\}_{i=0}^n \subseteq \mathcal{U}_\omega$ such that for any algorithm $A \in \mathcal{A}^B_{\{C_i\}_{i=0}^n}$ starting from $x^{(0)}$, the lower bound in* (3) *is also valid.*

Theorem 1 and Corollary 1 indicate that distributed learning with both unidirectional and bidirectional communication compression share the same lower bound. It is intuitive since unidirectional compression is just a special case of bidirectional compression by letting $C_0 = I$.

**3.3 Unidirectional Contractive Compressors.** To obtain the lower bounds for communication compression with contractive compressors, we need the following lemma [49, Lemma 1].

**Lemma 1 (Compressor relation).** *It holds that $\delta\,\mathcal{U}_{\delta^{-1}-1} \triangleq \{\delta\, C : C \in \mathcal{U}_{\delta^{-1}-1}\} \subseteq \mathcal{C}_\delta$.*

The above Lemma reveals that any $(\delta^{-1}-1)$-unbiased compressor is $\delta$-contractive when scaled by $\delta$. Therefore, if an algorithm $A$ admits all $\delta$-contractive compressors, it will also admits all compressors in $\delta\,\mathcal{U}_{\delta^{-1}-1}$ due to Lemma 1. This relation, together with Theorem 1, helps us achieve the following lower bound with respect to $\delta$-contractive compressors.

**Theorem 2 (Unidirectional contractive compression).** *For every $\Delta$, $L > 0$, $n \geq 2$, $0 < \delta \leq 1$, $\sigma > 0$, $T \geq \delta^{-2}$, there exists a set of loss objectives $\{f_i\}_{i=1}^n \subseteq \mathcal{F}_{\Delta,L}$, a set of stochastic gradient oracles $\{O_i\}_{i=1}^n \subseteq \mathcal{O}_{\sigma^2}$, a set of $\delta$-contractive compressors $\{C_i\}_{i=1}^n \subseteq \mathcal{C}_\delta$ with $C_0 = I$, such that for any algorithm $A \in \mathcal{A}^U_{\{C_i\}_{i=0}^n}$ starting from $x^{(0)}$, it holds that*

$$\mathbb{E}[\|\nabla f(\hat{x}_{A,\{f_i\}_{i=1}^n,\{O_i\}_{i=1}^n,\{C_i\}_{i=0}^n,T})\|^2] = \Omega\left(\left(\frac{\Delta L\sigma^2}{nT}\right)^{\frac{1}{2}} + \frac{\Delta L}{\delta T}\right). \tag{4}$$

**Transient complexity.** With the discussion on transient complexity in Sec. 3.2, it is easy to derive the transient iteration complexity as $\mathcal{O}(n/\delta^2)$ for the lower bound with $\delta$-contractive compressor.

**3.4 Bidirectional Contractive Compressors.** Noting that $\{C_i\}_{i=1}^n \subseteq \mathcal{C}_\delta$ with $C_0 = I$ is a special case of $\{C_i\}_{i=0}^n \subseteq \mathcal{C}_\delta$, we can also establish the lower bound for algorithms that admit bidirectional compression and contractive compressions.

**Corollary 2 (Bidirectional contractive compression).** *Under the same settings as in Theorem 2, there exists a set of loss objectives $\{f_i\}_{i=1}^n \subseteq \mathcal{F}_{\Delta,L}$, a set of stochastic gradient oracles $\{O_i\}_{i=1}^n \subseteq \mathcal{O}_{\sigma^2}$, a set of $\delta$-contractive compressors $\{C_i\}_{i=1}^n \subseteq \mathcal{C}_\delta$, such that for any algorithm $A \in \mathcal{A}^B_{\{C_i\}_{i=0}^n}$ starting form $x^{(0)}$, the lower bound* (4) *is also valid.*

**Algorithm 1** Fast Compressed Communication: $v^{(k,R)} = \text{FCC}(v^{(k,\star)}, C, R, \text{target receiver(s)})$

**Input:** The original vector $v^{(k,\star)}$ to communicate at iteration $k$; a compressor $C$; rounds $R$; initial vector $v^{(k,0)} = 0$; target receiver(s);
**for** $r = 0, \cdots, R-1$ **do**
  Compress $v^{(k,\star)} - v^{(k,r)}$ into $c^{(k,r)} = C(v^{(k,\star)} - v^{(k,r)})$
  Send $c^{(k,r)}$ to the target receiver(s)
  Update $v^{(k,r+1)} = v^{(k,r)} + c^{(k,r)}$
**end for** ▷ The set $\{c^{(k,r)}\}_{r=0}^{R-1}$ will be sent to the receiver during the for-loop
**return** Variable $v^{(k,R)}$. ▷ It holds that $v^{(k,R)} = \sum_{r=0}^{R-1} c^{(k,r)}$

# 4 NEOLITHIC: A nearly optimal algorithm

Comparing the best-known upper bounds listed in Table 1 with the established lower bounds in (3) and (4), we find existing algorithms may not be optimal. There exists a clear gap between their convergence rates and our established lower bounds. In this section, we propose NEOLITHIC to fill in this gap. Its rate will match the lower bounds established in (3) and (4) up to logarithm factors. NEOLITHIC can work with both unidirectional and bidirectional compressions, and it is compatible with both unbiased and contractive compressors. NEOLITHIC will be discussed in detail with bidirectional contractive compression in this section. It is easy to be adapted to other settings by simply removing the server-to-worker compression or utilizing the scaled unbiased compressor.

**4.1 Fast Compression Communication.** NEOLITHIC is built on a novel compression communication protocol listed in Algorithm 1, which we call fast compressed communication (FCC). Given an input vector $v^{(k,\star)}$ to communicate in the $k$-th iteration, FCC will first initialize $v^{(k,0)} = 0$ and then recursively compresses the residual with $c^{(k,r)} \triangleq C(v^{(k,\star)} - v^{(k,r)})$ and sends it to the receiver for $R$ consecutive rounds, see the main recursion in Algorithm 1. When FCC operation finishes, the sender will transmit a set of compressed variables $\{c^{(k,r)}\}_{r=0}^{R-1}$ to the receiver, and return $v^{(k,R)} = \sum_{r=0}^{R-1} c^{(k,r)}$ to itself. Quantity $v^{(k,R)}$ can be regarded as a compressed vector of the original input $v^{(k,\star)}$ after the FCC protocol.

The FCC protocol can be conducted by the server (for which all variables in FCC are without any subscripts, *e.g.,* $c^{(k,r)}$ and $v^{(k,r)}$) or by any worker $i$ (for which all variables are with a subscript $i$, *e.g.,* $c_i^{(k,r)}$ and $v_i^{(k,r)}$). When $R = 1$, FCC reduces to the standard compression utilized in existing literature [57, 53, 25, 62, 32]. While FCC requires $R$ rounds of communication per iteration, the following lemma establishes that the compression error will be exponentially decreased with the number of communication rounds $R$.

**Lemma 2 (FCC property).** *Let $C$ be a $\delta$-contractive compressor and $v^{(k,R)} = \text{FCC}(v^{(k,\star)}, C, R)$. It holds for any $R \geq 1$ and $v^{(k,\star)} \in \mathbb{R}^d$ that (proof is in Appendix B)*

$$\mathbb{E}[\|v^{(k,R)} - v^{(k,\star)}\|^2] \leq (1-\delta)^R \|v^{(k,\star)}\|^2, \quad \forall k = 0, 1, 2, \cdots. \tag{5}$$

When $R = 1$, the above FCC property (5) reduces to Assumption 4 for standard contractive compressors. When $R$ is large, FCC can output $v^{(k,R)}$ endowed with very small compression errors.

**4.2 NEOLITHIC Algorithm.** NEOLITHIC is described in Algorithm 2. The FCC protocol in NEOLITHIC communicates $R$ rounds per iteration. To balance the gradient queries and communication rounds, NEOLITHIC will query $R$ stochastic gradients per iteration, see the gradient accumulation step in Algorithm 2. Compared to other algorithms listed in Table 1, the proposed NEOLITHIC takes $R$ times more gradient queries and communication rounds than them per iteration. Given the same budgets to query stochastic gradients and conduct communications as the other algorithms, we shall consider $K = T/R$ iterations in NEOLITHIC for fair comparison.

We introduce the following assumption to establish the convergence rates of NEOLITHIC.

**Algorithm 2** NEOLITHIC

**Input:** Initialize $x^{(0)}$; learning rate $\gamma$; compression round $R$; $\delta^{(-1)} = \delta_i^{(-1)} = 0, \forall\, i \in [n]$
**for** $k = 0, 1, \cdots, K$ **do**
  **On all workers in parallel:**
    Query stochastic gradients $\hat{g}_i^{(k)} = \frac{1}{R}\sum_{r=0}^{R-1} O_i(x^{(k)}; \zeta_i^{(k,r)})$ ▷ Gradient accumulation
    Error compensate $\tilde{g}_i^{(k)} = \hat{g}_i^{(k)} + \delta_i^{(k-1)}$
    Update error $\delta_i^{(k)} = \tilde{g}_i^{(k)} - \text{FCC}(\tilde{g}_i^{(k)}, C_i, R, \text{server})$ ▷ Worker sends $\{c_i^{(k,r)}\}$ to server
  **On server:**
    Error compensate $\tilde{g}^{(k)} = \delta^{(k-1)} + \frac{1}{n}\sum_{i=1}^{n}\sum_{r=0}^{R-1} c_i^{(k,r)}$ ▷ $\{c_i^{(k,r)}\}$ received from workers
    Update error $\delta^{(k)} = \tilde{g}^{(k)} - \text{FCC}(\tilde{g}^{(k)}, C_0, R, \text{all workers})$ ▷ Server sends $\{c^{(k,r)}\}$ to workers
  **On all workers in parallel:**
    Update model parameter $x^{(k+1)} = x^{(k)} - \gamma\sum_{r=0}^{R-1} c^{(k,r)}$ ▷ $\{c^{(k,r)}\}$ received from server
**end for**

**Assumption 5 (Gradient dissimilarity).** *There exists some $b^2 \geq 0$ such that*

$$\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x) - \nabla f(x)\|^2 \leq b^2, \quad \forall\, x \in \mathbb{R}^d.$$

When local distributions $D_i$ are equivalent across all nodes $i$, we have $f_i(x) = f(x)$ and the above assumption will always hold. We first establish the convergence rate of NEOLITHIC with bidirectional and contractive compressors.

**Theorem 3 (NEOLITHIC with contractive compressors).** *Given constants $n \geq 1$, $\delta \in (0, 1]$ and Assumption 5, and let $x^{(k)}$ be generated by Algorithm 2. If $R = \lceil \frac{\max\{\ln(\delta T \max\{b^2, \sigma^2\delta\}/\Delta L), \ln(8)\}}{\delta} \rceil$ and learning rate is set as in Appendix B, it holds for any $K \geq 0$ and compressors $\{C_i\}_{i=0}^n \subseteq \mathcal{C}_\delta$ that*

$$\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}[\|\nabla f(x^{(k)})\|^2] = \tilde{O}\left(\left(\frac{\Delta L\sigma^2}{nT}\right)^{\frac{1}{2}} + \frac{\Delta L}{\delta T}\right),$$

*where $T = KR$ is the total number of gradient query/communication rounds on each worker and notation $\tilde{O}(\cdot)$ hides logarithmic factors. The above rate implies a transient complexity of $\tilde{\mathcal{O}}(n\delta^{-2})$.*

When the compressor $C$ is replaced with $\omega$-unbiased compressors, we utilize the fact that $(1+\omega)^{-1}C$ is $(1+\omega)^{-1}$-contractive to derive that

**Corollary 3 (NEOLITHIC with unbiased compressors).** *Under the same assumptions as in Theorem 3, it holds for any $K \geq 0$ and compressors $\{C_i\}_{i=0}^n \subseteq \mathcal{U}_\omega$ that*

$$\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}[\|\nabla f(x^{(k)})\|^2] = \tilde{O}\left(\left(\frac{\Delta L\sigma^2}{nT}\right)^{\frac{1}{2}} + \frac{(1+\omega)\Delta L}{T}\right).$$

*This further leads to a transient complexity of $\tilde{\mathcal{O}}\left(n(1+\omega)^2\right)$.*

**Remark 1.** *The convergence rates established in Theorem 3 and Corollary 3 are also valid for unidirectional compression when $C_0 = I$. They can match the lower bounds established in Sec. 3 up to logarithm factors. Moreover, these rates are faster than other algorithms listed in Table 1.*

**Remark 2.** *The results in Theorem 3 and Corollary 3 also imply that NEOLITHIC with bidirectional compression can perform as fast as its counterpart with unidirectional compression. In other words, imposing bidirectional compression can save communications in NEOLITHIC without hurting convergence rates. Before our work, it is established in literature [53, 50, 57, 64, 15] that bidirectional compression leads to slower convergence than unidirectional compression.*

**Remark 3.** *We remark that Assumption 5 is not required to obtain the lower bounds in Sec. 3. It is not known whether the lower bounds established in Sec. 3 can be achieved by NEOLITHIC when Assumption 5 does not hold. However, it is worth noting that Assumption 5 is milder than those made in most works [73, 34, 53, 62, 57, 12] such as bounded gradients. To our best knowledge, only*

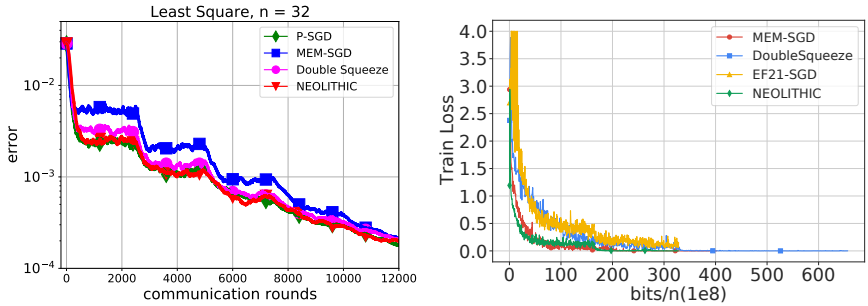


Figure 1: Left: Convergence on the synthetic least square problem in terms of $\|x - x^\star\|^2$ versus communication rounds. Right: Convergence on the CIFAR-10 in terms of training loss versus communication cost.

*EF21-SGD [25] is guaranteed to converge without Assumption 5, which, however, leads to a fairly loose convergence rate (see Table 1) that cannot show linear-speedup* $O(1/\sqrt{nT})$.

## 5 Experiments

This section empirically investigates the performance of different compression algorithms with both synthetic simulation and deep learning tasks. We compare NEOLITHIC with PSGD and its variants with communication compression: MEM-SGD, Double-Squeeze, and EF21-SGD. Implementation details and more experiments are provided in Appendix C.

**Least square.** We solve a synthetic least-square problem with all aforementioned algorithms. We set $d = 30$, $n = 32$, $R = 4$ (for NEOLITHIC) and utilize the rand-1 compressor. It is observed in Figure 1 (left) that NEOLITHIC beats MEM-SGD and Double Squeeze with performance close to P-SGD. It implies that algorithms with bidirectional compression converge no slower than the ones with unidirectional compression, which is consistent with results in Remark 1.

**Image classification.** We investigate the performance on two common ResNet models [28] with CIFAR-10 [37] dataset. We train total 300 epochs and set the batch size to 128 on every worker. All experiments were repeated three times with different seeds. For NEOLITHIC, we set $R = 2$. We utilize top-k compressor [57] with different compression ratios. As shown in Figure 1 (right) and Table 2, NEOLITHIC consistently outperforms other compression methods and reach the similar performance to PSGD.

Table 2: Accuracy comparison with different algorithms on CIFAR-10 (8 workers, 5% compression ratio).

| METHODS | RESNET18 | RESNET20 |
|---|---|---|
| PSGD | 93.99 ± 0.52 | 91.62 ± 0.13 |
| MEM-SGD | 94.35 ± 0.01 | 91.27 ± 0.08 |
| DOUBLE-SQUEEZE | 94.11 ± 0.14 | 90.73 ± 0.02 |
| EF-21 | 87.37 ± 0.49 | 65.82 ± 4.86 |
| NEOLITHIC | **94.63 ± 0.09** | **91.43 ± 0.10** |

**The effect of compression ratio.** We also investigate the influence of different compression ratios. Table 3 used a compression ratio of 1%, which indicates a more harsh setting for compression methods. NEOLITHIC still outperforms other compression methods.

Table 3: Accuracy comparison with different algorithms on CIFAR-10 (8 workers, 1% compression ratio).

| METHODS | RESNET18 | RESNET20 |
|---|---|---|
| MEM-SGD | 93.99 ± 0.11 | 89.68 ± 0.17 |
| DOUBLE-SQUEEZE | 93.54 ± 0.17 | 89.35 ± 0.04 |
| EF-21 | 67.78 ± 2.14 | 56.0 ± 2.257 |
| NEOLITHIC | **94.155 ± 0.10** | **89.82 ± 0.37** |

## 6 Conclusion

This paper provides lower bounds for distributed algorithms with communication compression, whether the compression is unidirectional or bidirectional and unbiased or contractive. An algorithm called NEOLITHIC is introduced to match the lower bounds under the assumption of bounded gradient dissimilarity. Future directions include developing optimal algorithms without the assumption, as well as discovering additional compression properties that might produce a better lower bound.

# References


[1] A. Agarwal and L. Bottou. A lower bound for the optimization of finite sums. In *International Conference on Machine Learning*, 2015.

[2] A. Agarwal and J. C. Duchi. Distributed delayed stochastic optimization. *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 5451–5452, 2012.

[3] S. A. Alghunaim and K. Yuan. A unified and refined convergence analysis for non-convex decentralized learning. *arXiv preprint arXiv:2110.09993*, 2021.

[4] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, 2017.

[5] D. Alistarh, T. Hoefler, M. Johansson, S. Khirirat, N. Konstantinov, and C. Renggli. The convergence of sparsified gradient methods. In *Advances in Neural Information Processing Systems*, 2018.

[6] Z. Allen-Zhu. How to make the gradients small stochastically: Even faster convex and nonconvex sgd. In *Advances in Neural Information Processing Systems*, 2018.

[7] Y. Arjevani, Y. Carmon, J. C. Duchi, D. J. Foster, N. Srebro, and B. Woodworth. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

[8] Y. Arjevani, Y. Carmon, J. C. Duchi, D. J. Foster, N. Srebro, and B. E. Woodworth. Lower bounds for non-convex stochastic optimization. *ArXiv*, abs/1912.02365, 2019.

[9] Y. Arjevani and O. Shamir. Communication complexity of distributed convex learning and optimization. In *Advances in Neural Information Processing Systems*, 2015.

[10] M. Assran, N. Loizou, N. Ballas, and M. Rabbat. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning (ICML)*, pages 344–353, 2019.

[11] E. Balkanski and Y. Singer. Parallelization does not accelerate convex optimization: Adaptivity lower bounds for non-smooth convex minimization. *ArXiv*, abs/1808.03880, 2018.

[12] D. Basu, D. Data, C. B. Karakuş, and S. N. Diggavi. Qsparse-local-sgd: Distributed sgd with quantization, sparsification, and local computations. *IEEE Journal on Selected Areas in Information Theory*, 1:217–226, 2020.

[13] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar. Signsgd: compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, 2018.

[14] J. Bernstein, J. Zhao, K. Azizzadenesheli, and A. Anandkumar. signsgd with majority vote is communication efficient and byzantine fault tolerant. *ArXiv*, abs/1810.05291, 2018.

[15] A. Beznosikov, S. Horvath, P. Richtárik, and M. H. Safaryan. On biased compression for distributed learning. *ArXiv*, abs/2002.12410, 2020.

[16] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*, 2010.

[17] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, pages 1–50, 2020.

[18] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Lower bounds for finding stationary points ii: first-order methods. *Mathematical Programming*, 185:315–355, 2021.

[19] J. Chen and A. H. Sayed. Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Transactions on Signal Processing*, 60(8):4289–4305, 2012.

[20] T. Chen, G. Giannakis, T. Sun, and W. Yin. LAG: Lazily aggregated gradient for communication-efficient distributed learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060, 2018.

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

[22] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. W. Senior, P. A. Tucker, K. Yang, and A. Ng. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, 2012.

[23] J. Diakonikolas and C. Guzmán. Lower bounds for parallel and randomized convex optimization. In *Conference on Learning Theory*, 2019.

[24] C. Fang, C. J. Li, Z. Lin, and T. Zhang. Spider: Near-optimal non-convex optimization via stochastic path integrated differential estimator. In *Advances in Neural Information Processing Systems*, 2018.

[25] I. Fatkhullin, I. Sokolov, E. A. Gorbunov, Z. Li, and P. Richtárik. Ef21 with bells & whistles: Practical algorithmic extensions of modern error feedback. *ArXiv*, abs/2110.03294, 2021.

[26] D. J. Foster, A. Sekhari, O. Shamir, N. Srebro, K. Sridharan, and B. E. Woodworth. The complexity of making the gradient small in stochastic convex optimization. In *Conference on Learning Theory*, 2019.

[27] E. A. Gorbunov, K. Burlachenko, Z. Li, and P. Richtárik. Marina: Faster non-convex distributed learning with compression. *ArXiv*, abs/2102.07845, 2021.

[28] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[29] S. Horvath, C.-Y. Ho, L. Horvath, A. N. Sahu, M. Canini, and P. Richtárik. Natural compression for distributed deep learning. *ArXiv*, abs/1905.10988, 2019.

[30] S. Horvath, D. Kovalev, K. Mishchenko, S. U. Stich, and P. Richtárik. Stochastic distributed learning with gradient quantization and variance reduction. *arXiv: Optimization and Control*, 2019.

[31] X. Huang, K. Yuan, X. Mao, and W. Yin. An improved analysis and rates for variance reduction under without-replacement sampling orders. In *Advances in Neural Information Processing Systems*, volume 34, pages 3232–3243, 2021.

[32] P. Jiang and G. Agrawal. A linear speedup analysis of distributed deep learning with sparse and quantized communication. In *Advances in Neural Information Processing Systems*, 2018.

[33] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning (ICML)*, pages 5132–5143. PMLR, 2020.

[34] S. P. Karimireddy, Q. Rebjock, S. U. Stich, and M. Jaggi. Error feedback fixes signsgd and other gradient compression schemes. In *International Conference on Machine Learning*, 2019.

[35] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[36] A. Koloskova, T. Lin, and S. U. Stich. An improved analysis of gradient tracking for decentralized machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[37] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[38] M. Li, D. G. Andersen, J. W. Park, A. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su. Scaling distributed machine learning with the parameter server. In *OSDI*, 2014.

[39] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 2017.

[40] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. *ArXiv*, abs/1712.01887, 2018.

[41] Y. Lu and C. D. Sa. Optimal complexity in decentralized training. In *International Conference on Machine Learning*, 2021.

[42] K. Mishchenko, E. A. Gorbunov, M. Takác, and P. Richtárik. Distributed learning with compressed gradient differences. *ArXiv*, abs/1901.09269, 2019.

[43] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[44] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o(1/k$\hat{2}$). In *Doklady an ussr*, volume 29, 1983.

[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035, 2019.

[46] S. Pu, A. Olshevsky, and I. C. Paschalidis. A sharp estimate on the transient time of distributed stochastic gradient descent. *arXiv preprint arXiv:1906.02702*, 2019.

[47] X. Qian, P. Richtárik, and T. Zhang. Error compensated distributed sgd can be accelerated. *arXiv*, 2020.

[48] P. Richtárik, I. Sokolov, and I. Fatkhullin. Ef21: A new, simpler, theoretically better, and practically faster error feedback. *ArXiv*, abs/2106.05203, 2021.

[49] M. H. Safaryan, E. Shulgin, and P. Richtárik. Uncertainty principle for communication compression in distributed and federated learning and the search for an optimal compressor. *Information and Inference: A Journal of the IMA*, 2021.

[50] A. N. Sahu, A. Dutta, A. M. Abdelmoniem, T. Banerjee, M. Canini, and P. Kalnis. Rethinking gradient sparsification as total error minimization. In *Advances in Neural Information Processing Systems*, 2021.

[51] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *INTERSPEECH*, 2014.

[52] S. U. Stich. Local sgd converges fast and communicates little. In *International Conference on Learning Representations (ICLR)*, 2019.

[53] S. U. Stich, J.-B. Cordonnier, and M. Jaggi. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems*, 2018.

[54] H. Sun, Y. Shao, J. Jiang, B. Cui, K. Lei, Y. Xu, and J. Wang. Sparse gradient compression for distributed sgd. In *Database Systems for Advanced Applications*, 2019.

[55] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu. Communication compression for decentralized training. In *Advances in Neural Information Processing Systems*, 2018.

[56] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu. $d^2$: Decentralized training over decentralized data. In *International Conference on Machine Learning*, pages 4848–4856, 2018.

[57] H. Tang, X. Lian, T. Zhang, and J. Liu. Doublesqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. *ArXiv*, abs/1905.05957, 2019.

[58] T. Vogels, S. P. Karimireddy, and M. Jaggi. Powersgd: Practical low-rank gradient compression for distributed optimization. In *Advances in Neural Information Processing Systems*, 2019.

[59] J. Wangni, J. Wang, J. Liu, and T. Zhang. Gradient sparsification for communication-efficient distributed optimization. In *Advances in Neural Information Processing Systems*, 2018.

[60] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. H. Li. Terngrad: Ternary gradients to reduce communication in distributed deep learning. In *Advances in Neural Information Processing Systems*, 2017.

[61] J. Wu, W. Huang, J. Huang, and T. Zhang. Error compensated quantized sgd and its applications to large-scale distributed optimization. In *International Conference on Machine Learning*, 2018.

[62] C. Xie, S. Zheng, O. Koyejo, I. Gupta, M. Li, and H. Lin. Cser: Communication-efficient sgd with error reset. In *Advances in Neural Information Processing Systems*, 2020.

[63] R. Xin, U. A. Khan, and S. Kar. An improved convergence analysis for decentralized online stochastic non-convex optimization. *IEEE Transactions on Signal Processing*, 2020.

[64] A. Xu, Z. Huo, and H. Huang. Step-ahead error feedback for distributed training with compressed gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[65] H. Xu, C.-Y. Ho, A. M. Abdelmoniem, A. Dutta, E. H. Bergou, K. Karatsenidis, M. Canini, and P. Kalnis. Compressed communication for distributed deep learning: Survey and quantitative evaluation. *Technical report*, 2020.

[66] B. Ying, K. Yuan, Y. Chen, H. Hu, P. Pan, and W. Yin. Exponential graph is provably efficient for decentralized deep training. *Advances in Neural Information Processing Systems (NeurIPS), 34. Also available at arXiv:2110.13363*, 2021.

[67] 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*, pages 7184–7193. PMLR, 2019.

[68] K. Yuan, S. A. Alghunaim, B. Ying, and A. H. Sayed. On the influence of bias-correction on distributed stochastic optimization. *IEEE Transactions on Signal Processing*, 2020.

[69] K. Yuan, Q. Ling, and W. Yin. On the convergence of decentralized gradient descent. *SIAM Journal of Optimization*, 26:1835–1854, 2016.

[70] N. Yurri. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, Norwell, 2004.

[71] M. D. Zeiler. Adadelta: An adaptive learning rate method. *ArXiv*, abs/1212.5701, 2012.

[72] H. Zhang, J. Li, K. Kara, D. Alistarh, J. Liu, and C. Zhang. Zipml: Training linear models with end-to-end low precision, and a little bit of deep learning. In *International Conference on Machine Learning*, 2017.

[73] S.-Y. Zhao, Y.-P. Xie, H. Gao, and W.-J. Li. Global momentum compression for sparse communication in distributed sgd. *ArXiv*, abs/1905.12948, 2019.

[74] D. Zhou and Q. Gu. Lower bounds for smooth nonconvex finite-sum optimization. *ArXiv*, abs/1901.11224, 2019.

# Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]