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

Xinmeng Huang[1*] **Yiming Chen**[2,3*] **Wotao Yin**[2] **Kun Yuan**[2,4]

[1]University of Pennsylvania [2]DAMO Academy, Alibaba Group
[3]MetaCarbon [4]Peking University

xinmengh@sas.upenn.edu yiming@metacarbon.vip
{wotao.yin, kun.yuan}@alibaba-inc.com

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, 12, 54], decentralized communication [37, 9, 19], lazy communication [71, 55, 31, 18], and beyond. This article will focus on the former.

The most common method of distributed training is Parallel SGD (P-SGD) [15]. 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, 27, 4] or compress the messages between them bidirectionally [60, 68]. The method of compression is either sparsification or quantization [5, 27, 4] or their combination [27, 14]. The literature [56, 53, 60, 68] reveals that bidirectional compression can save more communication, but leads to slower convergence rates.

*Equal Contribution. Corresponding Author: Kun Yuan

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

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 σ^2, data heterogeneity bound b^2 , gradient square norm bound G^2 (used in [56, 60, 66], 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 $\tilde{O}(\cdot)$ hides all logarithmic terms. "LB" and "UB" indicate lower and upper bound, respectively. Notation δ and ω are compressor-related parameters (see detailed discussions in Section 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	$\Omega(n/\delta^2)$
	Theorem 2+Corollary 2	$\Omega\left(\frac{\sigma}{\sqrt{nT}} + \frac{1+\omega}{T}\right)$	Uni/Bidirectional Unbiased	$\Omega\left(n(1+\omega)^2\right)$
UB	Theorem 3	$\tilde{O}\left(\frac{\sigma}{\sqrt{nT}} + \frac{1}{\delta T}\right)$	Uni/Bidirectional Contractive	$\tilde{O}(n/\delta^2)$
	Corollary 3	$\tilde{O}\left(\frac{\sigma}{\sqrt{nT}} + \frac{1+\omega}{T}\right)$	Uni/Bidirectional Unbiased	$\tilde{O}(n(1+\omega)^2)$
	Q-SGD [30]	$O\left(\frac{(1+\omega)^{1/2}\sigma+\omega^{1/2}b}{\sqrt{nT}}\right)^{\dagger,\diamond}$	Unidirectional i.i.d, Unbiased	–
	MEM-SGD [56]	$O\left(\frac{\sigma}{\sqrt{nT}} + \frac{G^{2/3}}{\delta^{2/3}T^{2/3}} + \frac{1}{T}\right)^{\diamond}$	Unidirectional Contractive	$O(n^3/\delta^4)$
	Double-Squeeze [60]	$O\left(\frac{1}{\sqrt{nT}} + \frac{G^{2/3}}{\delta^{4/3}T^{2/3}} + \frac{1}{T}\right)^{\diamond}$	Unidirectional Contractive	$O(n^3/\delta^8)$
	CSER [66]	$O\left(\frac{\sigma}{\sqrt{nT}} + \frac{G^{2/3}}{\delta^{2/3}T^{2/3}} + \frac{1}{T}\right)^{\diamond}$	Unidirectional Contractive	$O(n^3/\delta^4)$
	EF21-SGD [23]	$O\left(\frac{\sigma}{\sqrt{\delta^3 T}} + \frac{1}{\delta T}\right)^{*}$	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.*, $O(\frac{\sigma}{\sqrt{nT}})$.
† This convergence rate is valid only for $T = \Omega(n(1+\frac{\omega}{n})^2)$. Since the rate $\frac{(1+\omega)^{1/2}\sigma+\omega^{1/2}b}{\sqrt{nT}}$ 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.
◇ The rates are either strengthened by relaxing the original restrictive assumptions or extended to the same setting as NEOLITHIC for a fair comparison, see more details in Appendix D.

Although there are many specific compression methods, their convergence analyses are mainly built on unbiased compressibility or contractive compressibility. The literature [14, 52, 69] 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 [14, 42, 28, 25].

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, for non-convex problems, algorithms using biased contractive compressors bidirectionally can theoretically converge as fast as those with unbiased compressors used unidirectionally. Before out work, it is only established in [46] that, for convex problems and unbiased compressors, algorithms with bidirectional compression can match with the counterparts with unidirectional compression.
- 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 Section 3), is also listed in the table. The smaller the transient complexity is, the faster the algorithm converges.

1.2 Related Works

Distributed learning. Distributed learning has been increasingly useful in training large-scale machine learning models [20]. It typically follows a centralized or decentralized setup. Centralized approaches [2, 36], 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, 75, 74, 37], D^2[59, 72], stochastic gradient tracking [67, 34, 3], and their momentum variants [38, 73]. The lazy communication [71, 55, 18] 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 [30, 58, 78] is essentially an unbiased operator with random noise. For example, [54] develops Sign-SGD by using only 1 bit for each entry whose convergence is studied in [12, 13, 64]. 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. [63, 56] 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 [62, 39, 57] due to analysis challenges, and they are established with assumptions such as bounded gradients [80, 32] or quadratic loss functions [65]. More discussions on unbiased and biased compressors can be found in [14, 52, 51]. Communication compression can also be combined with other communication-saving techniques such as decentralization [40, 79].

Error compensation. The error compensation (feedback) mechanism is introduced by [54] to mitigate the error caused by 1-bit quantization. [65] studies SGD with error-compensated quantization for quadratic functions with convergence guarantees. [56] 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 [50] 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 [23] yet no linear-speedup rate is achieved.

Lower bounds in optimization. Lower bounds are well studied in convex optimization [22, 1, 21] especially when there is no gradient noise [8, 76, 10, 6, 24]. In non-convex optimization, [16, 17] 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 [81, 7] extend the analysis technique to finite sum and stochastic problems. Recently, [41, 74] also show the lower bounds in the decentralized stochastic setting based on a similar approach. For distributed learning with communication compression, a useful work [47] establishes a lower bound for unidirectional/bidirectional compression for strongly convex problems with fixed learning rates and unbiased compressors. There are limited studies on lower bounds for non-convex distributed learning with unbiased/contractive compressions.

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 ξ. 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 ζ_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 ω-*unbiased* compressor, described by Assumption 3, *e.g.*, the stochastic quantization operator [4], and ii) the δ-*contractive* compressor, described by Assumption 4, *e.g.*, the rand-K and top-K operators [56, 49].

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 ω-unbiased compressors and δ-contractive compressors satisfying Assumptions 3 and 4, respectively. Note that the identity operator I satisfies $I \in \mathcal{U}_\omega$ for any $\omega \geq 0$ and $I \in \mathcal{C}_\delta$ for any $\delta \in (0, 1]$. Generally, an ω-unbiased compressor is not necessarily contractive when ω 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,

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

We further require algorithms A to satisfy the so-called "zero-respecting" property, which appears in [16, 17, 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, 33, 29, 77]. 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, $\forall\, 1 \leq i \leq n$, 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 ω or δ to be accessible by algorithm A.

3.1 Unidirectional Unbiased Compression

Our first result is for algorithms that admit unidirectional compression and ω-unbiased compressors.

Theorem 1 (**Unidirectional unbiased compression**). *For every Δ, $L > 0$, $n \geq 2$, $\omega \geq 0$, $\sigma > 0$, $T \geq (1+\omega)^2$, 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 [7]. 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 [16].

Linear-speedup. When T is sufficiently large, the first term $1/\sqrt{nT}$ dominates the lower bound (3). If an algorithm achieves an $O(1/\sqrt{nT})$ rate, it will require $T = O(1/(n\epsilon^2))$ gradient queries to reach a desired accuracy ϵ, which is inversely proportional to n. Therefore, an algorithm achieves linear-speedup at Tth 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., $T = \Omega(n(1+\omega)^2)$ 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 [48, 70] to gauge how the network topology can influence the convergence rate.

3.2 Bidirectional Unbiased Compression

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 Compression

To obtain the lower bounds for communication compression with contractive compressors, we need the following lemma [52, 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 δ-contractive when scaled by δ. Therefore, if an algorithm A admits all δ-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 δ-contractive compressors.

Theorem 2 (**Unidirectional contractive compression**). *For every* Δ, $L > 0$, $n \geq 2$, $0 < \delta \leq 1$, $\sigma > 0$, $T \geq \delta^{-2}$, *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 δ-contractive compressors* $\{C_i\}_{i=1}^n \subseteq \mathcal{C}_\delta$ *with* $C_0 = I$, *such that*

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 $v^{(k,R)} = \sum_{r=0}^{R-1} c^{(k,r)}$

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 Section 3.1, it is easy to derive the transient iteration complexity as $\Omega(n/\delta^2)$ for the lower bound with δ-contractive compressor.

3.4 Bidirectional Contractive Compression.

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 δ-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.*

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 Compressed Communication

NEOLITHIC is built on a 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 [60, 56, 23, 66, 30]. 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.

Algorithm 2 NEOLITHIC

Input: Initialize $x^{(0)}$; learning rate γ; 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 $\delta_i^{(k)} = \tilde{g}_i^{(k)} - \mathrm{FCC}(\tilde{g}_i^{(k)}, C_i, R, \text{server})$ ▷ 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 $\delta^{(k)} = \tilde{g}^{(k)} - \mathrm{FCC}(\tilde{g}^{(k)}, C_0, R, \text{all workers})$ ▷ 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

Lemma 2 (**FCC property**). *Let C be a δ-contractive compressor and $v^{(k,R)} = \mathrm{FCC}(v^{(k,\star)}, C, R)$. 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. The FCC protocol is also closely related to EF21 compression strategy [50][2]. However, the major novelty of FCC is to utilize multiple such compression rounds to help develop algorithms that can nearly match the established lower bounds.

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 standard compression 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.

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 $R = \lceil \max\{\ln\left(\delta T \max\{b^2, \sigma^2\delta\}/\Delta L\right), \ln(8)\}/\delta \rceil$ 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 $\tilde{O}(n\delta^{-2})$.

[2] If the roles of v and $v^\star$ in Eq. (8) of [50] are switched, we can get the single round of FCC recursions.

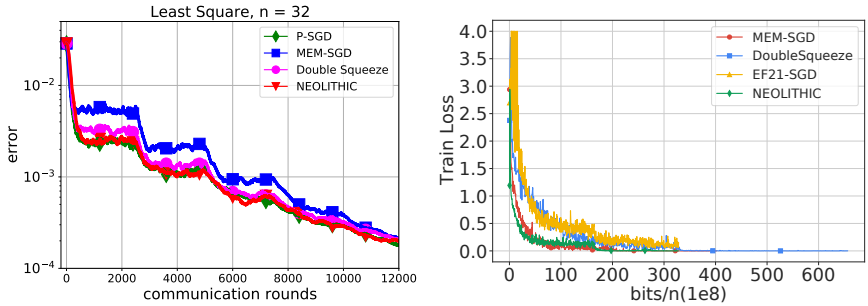

Figure 1: Left: Convergence on the synthetic least square problem in terms of $\mathbb{E}[\|\nabla f(x)\|^2]$ versus communication rounds. Right: Convergence on the CIFAR-10 in terms of training loss versus communication cost.

When the compressor C is replaced with ω-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 $\tilde{O}\left(n(1+\omega)^2\right)$.

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 Section 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 [56, 53, 60, 68, 14] that bidirectional compression leads to slower convergence than unidirectional compression for non-convex problems.*

Remark 3. *We remark that Assumption 5 is not required to obtain the lower bounds in Section 3. It is not known whether the lower bounds established in Section 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 [80, 32, 56, 66, 60, 11] such as bounded gradients. To our best knowledge, only EF21-SGD [23] is guaranteed to converge without Assumption 5, which, however, leads to a fairly loose convergence rate (see Table 1) that cannot show linear-speedup $O(1/\sqrt{nT})$.*

Remark 4. *In the deterministic scenario, i.e., $\sigma^2 = 0$, NEOLITHIC produces rate $\tilde{O}((1+\omega)\Delta L/T)$. There exist literature (e.g., [25, 61]) that can achieve rate $\tilde{O}((1+\omega/n)\Delta L/T)$ which outperforms NEOLITHIC especially when ω and n are sufficiently large. However, this improvement is built upon an additional assumption that all local compressors $\{C_i\}_{i=0}^n$ are independent of each other and cannot share the same randomness. On the contrary, NEOLITHIC does not impose such an assumption and can be applied to both dependent and independent compressors.*

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.

Linear regression. 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 can converge as fast as the counterparts with unidirectional compression, which is consistent with results in Remark 1.

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**

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**

Image classification. We investigate the performance on two common ResNet models [26] with CIFAR-10 [35] 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 [60] 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.

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.

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.

Acknowledgements

The authors are grateful to Professor Peter Richtarik from KAUST for the helpful discussions in the relation between FCC and EF21 and various other useful suggestions.

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. In *IEEE Conference on Decision and Control*, 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. E. Woodworth. Lower bounds for non-convex stochastic optimization. *ArXiv*, abs/1912.02365, 2019.

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

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

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

[11] 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.

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

[13] 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.

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

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

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

[17] 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.

[18] 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.

[19] 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*, 2021.

[20] 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.

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

[22] 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.

[23] 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.

[24] 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.

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

[26] 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.

[27] 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.

[28] 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.

[29] X. Huang, K. Yuan, X. Mao, and W. Yin. Improved analysis and rates for variance reduction under without-replacement sampling orders. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[30] 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.

[31] 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*, 2020.

[32] 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.

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

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

[35] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[36] 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.

[37] 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.

[38] T. Lin, S. P. Karimireddy, S. U. Stich, and M. Jaggi. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. In *International Conference on Machine Learning*, 2021.

[39] 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.

[40] X. Liu, Y. Li, R. Wang, J. Tang, and M. Yan. Linear convergent decentralized optimization with compression. In *International Conference on Learning Representations*, 2021.

[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/k2). 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*, pages 8024–8035, 2019.

[46] C. Philippenko and A. Dieuleveut. Preserved central model for faster bidirectional compression in distributed settings. In *Advances in Neural Information Processing Systems*, 2021.

[47] C. Philippenko and A. Dieuleveut. Bidirectional compression in heterogeneous settings for distributed or federated learning with partial participation: tight convergence guarantees. *ArXiv*, 2022.

[48] 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.

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

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

[51] P. Richt'arik, I. Sokolov, I. Fatkhullin, E. Gasanov, Z. Li, and E. A. Gorbunov. 3pc: Three point compressors for communication-efficient distributed training and a better theory for lazy aggregation. In *International Conference on Machine Learning*, 2022.

[52] 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.

[53] 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.

[54] 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.

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

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

[57] 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.

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

[59] 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.

[60] 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.

[61] A. Tyurin and P. Richt'arik. Dasha: Distributed nonconvex optimization with communication compression, optimal oracle complexity, and no client synchronization. *ArXiv*, 2022.

[62] 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.

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

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] 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.

[70] B. Ying, K. Yuan, Y. Chen, H. Hu, P. Pan, and W. Yin. Exponential graph is provably efficient for decentralized deep training. In *Advances in Neural Information Processing Systems*, 2021.

[71] 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.

[72] K. Yuan, S. A. Alghunaim, and X. Huang. Removing data heterogeneity influence enhances network topology dependence of decentralized sgd. *arXiv preprint arXiv:2105.08023*, 2021.

[73] K. Yuan, Y. Chen, X. Huang, Y. Zhang, P. Pan, Y. Xu, and W. Yin. Decentlam: Decentralized momentum sgd for large-batch deep training. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3009–3019, 2021.

[74] K. Yuan, X. Huang, Y. Chen, X. Zhang, Y. Zhang, and P. Pan. Revisiting optimal convergence rate for smooth and non-convex stochastic decentralized optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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

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

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

[78] 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.

[79] H. Zhao, B. Li, Z. Li, P. Richt'arik, and Y. Chi. Beer: Fast o(1/t) rate for decentralized nonconvex optimization with communication compression. In *Advances in Neural Information Processing Systems*, 2022.

[80] 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.

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

Checklist

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]

A Lower Bounds

In this section, we provide the proofs for Theorem 1 and 2.

A.1 Proof of Theorem 1

Without loss of generality, we assume algorithms to start from $x^{(0)} = 0$. We denote the j-th coordinate of a vector $x \in \mathbb{R}^d$ by $[x]_j$ for $j = 1, \dots, d$, and let $\mathrm{prog}(x)$ be

$$\mathrm{prog}(x) := \begin{cases} 0 & \text{if } x = 0; \\ \max_{1 \le j \le d}\{j : [x]_j \neq 0\} & \text{otherwise.} \end{cases}$$

Similarly, for a set of points $\mathcal{X} = \{x_1, x_2, \dots\}$, we define $\mathrm{prog}(\mathcal{X}) := \max_{x \in \mathcal{X}} \mathrm{prog}(x)$. As described in [16, 17], a zero chain function f satisfies

$$\mathrm{prog}(\nabla f(x)) \le \mathrm{prog}(x) + 1, \quad \forall\, x \in \mathbb{R}^d,$$

which implies that, starting from $x = 0$, a single gradient evaluation can only make at most one more coordinate for the model parameter x be non-zero.

We prove the two terms of the lower bound in Theorem 1 separately by constructing two hard-to-optimize examples. The construction of the example, for each term in (3), can be conducted in three steps: 1) constructing local functions $\{f_i\}_{i=1}^n$ by following the zero-chain function model proposed by [16]; 2) constructing compressors $\{C_i\}_{i=1}^n \in \mathcal{U}_\omega$ and oracles $\{O_i\}_{i=1}^n \subseteq \mathcal{O}_{\sigma^2}$ that hamper algorithms to increase the non-zero coordinates of the model parameters; 3) show a limitation in terms of the non-zero coordinates of model parameters, for algorithms obeying the desired protocol with T gradient queries and compressed communications on each worker, and translate the limitation into the lower bound of convergence rate.

We first state some key zero-chain functions that will be used to facilitate the analysis.

Lemma 3 (Lemma 2 of [7])**.** *Let $[x]_j$ denote the j-th coordinate of a vector $x \in \mathbb{R}^d$, and define function*

$$h(x) := -\psi(1)\phi([x]_1) + \sum_{j=1}^{d-1} \Big(\psi(-[x]_j)\phi(-[x]_{j+1}) - \psi([x]_j)\phi([x]_{j+1}) \Big)$$

where for $\forall\, z \in \mathbb{R}$,

$$\psi(z) = \begin{cases} 0 & z \le 1/2; \\ \exp\left(1 - \frac{1}{(2z-1)^2}\right) & z > 1/2, \end{cases} \quad \phi(z) = \sqrt{e} \int_{-\infty}^{z} e^{\frac{1}{2}t^2} \mathrm{d}t.$$

Then h satisfy the following properties:

1. *$h(x) - \inf_x h(x) \le \delta_0 d$, $\forall\, x \in \mathbb{R}^d$ with $\delta_0 = 12$.*
2. *h is ℓ_0-smooth with $\ell_0 = 152$.*
3. *$\|\nabla h(x)\|_\infty \le g_\infty$, $\forall\, x \in \mathbb{R}^d$ with $g_\infty = 23$.*
4. *$\|\nabla h(x)\|_\infty \ge 1$ for any $x \in \mathbb{R}^d$ with $[x]_d = 0$.*

Lemma 4. *Let functions*

$$h_1(x) := -2\psi(1)\phi([x]_1) + 2 \sum_{j \text{ even},\, 0<j<d} \Big(\psi(-[x]_j)\phi(-[x]_{j+1}) - \psi([x]_j)\phi([x]_{j+1}) \Big)$$

and

$$h_2(x) := 2 \sum_{j \text{ odd},\, 0<j<d} \Big(\psi(-[x]_j)\phi(-[x]_{j+1}) - \psi([x]_j)\phi([x]_{j+1}) \Big).$$

Then h_1 and h_2 satisfy the following properties:

1. *$\frac{1}{2}(h_1 + h_2) = h$, where h is defined in Lemma 3.*

2. *For any* $x \in \mathbb{R}^d$*, if* $\mathrm{prog}(x)$ *is odd, then* $\mathrm{prog}(\nabla h_1(x)) \leq \mathrm{prog}(x)$*; if* $\mathrm{prog}(x)$ *is even, then* $\mathrm{prog}(\nabla h_2(x)) \leq \mathrm{prog}(x)$.

3. h_1 *and* h_2 *are also* ℓ_0*-smooth with* $\ell_0 = 152$.

Proof of Lemma 4. The first property follows the definition of h_1, h_2, and h. The second property follows Lemma 1 of [16] that $\psi^{(m)}(0) = 0$ for any $m \in \mathbb{N}$. Now we prove the third property. Note that the Hessian of h_k for $k = 1, 2$ is tridiagonal and symmetric. Consequently, by the Schur test, we have for any $x \in \mathbb{R}^d$ and $k = 1, 2$

$$\|\nabla^2 h_k(x)\|_2 \leq \sqrt{\|\nabla^2 h_k(x)\|_1 \|\nabla^2 h_k(x)\|_\infty} = \|\nabla^2 h_k(x)\|_1. \tag{6}$$

Furthermore, it is easy to verify that

$$\begin{aligned} &\|\nabla^2 h_k(x)\|_1 \\ \leq &2 \max\left\{\sup_{z\in\mathbb{R}} |\psi''(z)| \sup_{z\in\mathbb{R}} |\phi(z)|, \sup_{z\in\mathbb{R}} |\psi(z)| \sup_{z\in\mathbb{R}} |\phi''(z)|\right\} + 2 \sup_{z\in\mathbb{R}} |\psi'(z)| \sup_{z\in\mathbb{R}} |\phi'(z)| \leq 152, \end{aligned} \tag{7}$$

where the last inequality follows Observation 2 of [7] that

$$0 \leq \psi \leq e, \;\; 0 \leq \psi' \leq \sqrt{54/e}, \;\; |\psi''| \leq 32.5, \;\; 0 \leq \phi \leq \sqrt{2\pi e}, \;\; 0 \leq \phi' \leq \sqrt{e} \text{ and } |\phi''| \leq 1.$$

Combining (7) with (6), we know h_k is ℓ_0-smooth for $k = 1, 2$. □

Given Lemmas 3 and 4, we now construct two examples that lead to the two terms in lower bound (3), respectively.

Example 1. The proof of the first term $\Omega((\frac{\Delta L \sigma^2}{nT})^{\frac{1}{2}})$ essentially follows the first example in proving Theorem 1 of [41]. We provide the proof for the sake of being self-contained.

(Step 1.) Let $f_i = L\lambda^2 h(x/\lambda)/\ell_0, \forall\, i = 1, \ldots, n$ be homogeneous and hence $f = L\lambda^2 h(x/\lambda)/\ell_0$ where h is defined in Lemma 3 and $\lambda > 0$ is to be specified. Since $\nabla^2 f_i = L\nabla^2 h/\ell_0$ and h is ℓ_0-smooth (Lemma 3), we know f_i is L-smooth for any $\lambda > 0$. By Lemma 3, we have

$$f(0) - \inf_x f(x) = \frac{L\lambda^2}{\ell_0^2}(h(0) - \inf_x h(x)) \leq \frac{L\lambda^2 \delta_0 d}{\ell_0}.$$

Therefore, to ensure $f_i \in \mathcal{F}_{\Delta,L}$, it suffices to let

$$\frac{L\lambda^2 \delta_0 d}{\ell_0} \leq \Delta, \quad \text{i.e.,} \quad d\lambda^2 \leq \frac{\ell_0 \Delta}{L\delta_0}. \tag{8}$$

(Step 2.) We consider all compressors $\{C_i\}_{i=1}^n$ are identity, meaning that there is no compression error in the optimization procedure. Then we naturally have $\{C_i\}_{i=1}^n \subseteq \mathcal{U}^\omega$ for any $\omega \geq 0$. We construct the stochastic gradient oracle O_i on worker $i, \forall\, i = 1, \ldots, n$ as the following:

$$[O_i(x; Z)]_j = [\nabla f_i(x)]_j \left(1 + \mathbb{1}\{j > \mathrm{prog}(x)\}\left(\frac{Z}{p} - 1\right)\right), \forall\, x \in \mathbb{R}^d,\, j = 1, \ldots, d$$

with random variable $Z \sim$ Bernoulli(p) independent of x and f_i, and $p \in (0, 1)$ to be specified. The oracle O_i has probability to make the entry $[\nabla f_i(x)]_{\mathrm{prog}(x)+1}$ zero. Therefore algorithms are hampered by the gradient oracle to achieve more non-zero coordinates for the model parameters. It is easy to see O_i is an unbiased stochastic gradient oracle. Moreover, since f_i is zero-chain, we have $\mathrm{prog}(O_i(x; Z)) \leq \mathrm{prog}(\nabla f_i(x)) \leq \mathrm{prog}(x) + 1$ and hence

$$\begin{aligned} \mathbb{E}[\|[O_i(x; Z)] - \nabla f_i(x)\|^2] &= |[\nabla f_i(x)]_{\mathrm{prog}(x)+1}|^2 \mathbb{E}\left[\left(\frac{Z}{p} - 1\right)^2\right] = |[\nabla f_i(x)]_{\mathrm{prog}(x)+1}|^2 \frac{1-p}{p} \\ &\leq \|\nabla f_i(x)\|_\infty^2 \frac{1-p}{p} \leq \frac{L^2\lambda^2(1-p)}{\ell_0^2 p} \|\nabla h(x)\|_\infty^2 \\ &\overset{\text{Lemma 3}}{\leq} \frac{L^2\lambda^2(1-p) g_\infty^2}{\ell_0^2 p}. \end{aligned}$$

Therefore, to ensure $O_i \in \mathcal{O}_{\sigma^2}$, it suffices to let

$$p = \min\{\frac{L^2\lambda^2 g_\infty^2}{\ell_0^2\sigma^2}, 1\}. \tag{9}$$

(Step 3.) Let $x_i^{(t)}, \forall\, t = 0, \dots$ and $1 \le i \le n$, be the t-th query point of worker i. Since algorithms satisfy the zero-respecting property, as discussed in [16, 17, 41], within T gradient queries on each worker, algorithms can only return model $\hat{x}$ such that

$$\hat{x} \in \operatorname{span}\left(\left\{x^{(0)}, \nabla f_i(x^{(0)}), \left\{\{x_i^{(t)}, \nabla f_i(x_i^{(t)}) : 0 \le t < T\} : 1 \le i \le n\right\}\right\}\right),$$

which implies

$$\operatorname{prog}(\hat{x}) \le \max_{0\le t<T} \max_{1\le i\le n} \operatorname{prog}(x_i^{(t)}) + 1. \tag{10}$$

By Lemma 2 of [41], we have

$$\mathbb{P}(\operatorname{prog}(\hat{x}) \ge d) \le \mathbb{P}(\max_{0\le t<T} \max_{1\le i\le n} \operatorname{prog}(x_i^{(t)}) \ge d-1) \le e^{(e-1)npT-d+1}. \tag{11}$$

On the other hand, when $\operatorname{prog}(\hat{x}) < d$, by the fourth point in Lemma 3, it holds that

$$\min_{\hat{x}\in\operatorname{span}\{\{x_i^{(t)}:1\le i\le n,\, 0\le t<T\}\}} \|\nabla f(\hat{x})\| \ge \min_{[\hat{x}]_d=0} \|\nabla f(\hat{x})\| = \frac{L\lambda}{\ell_0} \min_{[\hat{x}]_d=0} \|\nabla h(\hat{x})\| \ge \frac{L\lambda}{\ell_0}. \tag{12}$$

Therefore, by combining (11) and (12), we have

$$\mathbb{E}[\|\nabla f(\hat{x})\|^2] \ge \mathbb{P}(\operatorname{prog}^{(T)} < d)\mathbb{E}[\|\nabla f(\hat{x})\|^2 \mid \operatorname{prog}^{(T)} < d] \ge (1 - e^{(e-1)npT-d+1})\frac{L^2\lambda^2}{\ell_0^2}. \tag{13}$$

Let

$$\lambda = \frac{\ell_0}{L}\left(\frac{\Delta L\sigma^2}{3nT\ell_0\delta_0 g_\infty^2}\right)^{\frac{1}{4}} \quad \text{and} \quad d = \left\lfloor \left(\frac{3L\Delta nTg_\infty^2}{\sigma^2\ell_0\delta_0}\right)^{\frac{1}{2}} \right\rfloor. \tag{14}$$

Then (8) naturally holds and $p = \min\{\frac{g_\infty^2}{\sigma^2}\left(\frac{\Delta L\sigma^2}{3nT\ell_0\delta_0 g_\infty^2}\right)^{\frac{1}{2}}, 1\}$ by (9). Without loss of generality, we assume $d \ge 2$, which is guaranteed when $T = \Omega(\frac{\sigma^2}{nL\Delta})$. Then, using the definition of p, we have that

$$\begin{aligned}
&(e-1)npT - d + 1 \le (e-1)nT\,\frac{g_\infty^2}{\sigma^2}\left(\frac{\Delta L\sigma^2}{3nT\ell_0\delta_0 g_\infty^2}\right)^{\frac{1}{2}} - d + 1 \\
=&\frac{e-1}{3}\left(\frac{3L\Delta nTg_\infty^2}{\sigma^2\ell_0\delta_0}\right)^{\frac{1}{2}} - d + 1 < \frac{e-1}{3}(d+1) - d + 1 \le 2 - e < 0
\end{aligned}$$

which, combined with (13), further implies

$$\mathbb{E}[\|\nabla f(\hat{x})\|^2] = \Omega\left(\frac{L^2\lambda^2}{\ell_0^2}\right) = \Omega\left(\left(\frac{\Delta L\sigma^2}{3nT\ell_0\delta_0 g_\infty^2}\right)^{\frac{1}{2}}\right) = \Omega\left(\left(\frac{\Delta L\sigma^2}{nT}\right)^{\frac{1}{2}}\right).$$

Example 2. Without loss of generality, we assume n is even, otherwise we can consider the lower bound for the case of $n-1$.

(Step 1.) Similar to the construction of Example 1, we let $f_i = L\lambda^2 h_1(x/\lambda)/\ell_0, \forall\, 1 \le i \le n/2$ and $f_i = L\lambda^2 h_2(x/\lambda)/\ell_0, \forall\, n/2 < i \le n$, where h_1 and h_2 are defined in Lemma 4, and $\lambda > 0$ will be specified later. By the definitions of h_1 and h_2, we have that $f_i, \forall\, 1 \le i \le n$ is zero-chain and $f(x) = \frac{1}{n}\sum_{i=1}^n f_i(x) = L\lambda^2/\ell_0 h(x/\lambda)$. Since h_1 and h_2 are also ℓ_0-smooth, to let $f \in \mathcal{F}_{\Delta,L}$, it suffices to make (8) hold.

In the above construction, we essentially split a zero-chain function, i.e., h, into two different components: the even component of the chain, i.e., h_1 and the odd component of the chain, i.e., h_2. Recall the second property in Lemma 4 that for any $x \in \mathbb{R}^d$, if $\operatorname{prog}(x)$ is odd, then $\operatorname{prog}(\nabla h_1(x)) \le \operatorname{prog}(x)$; if $\operatorname{prog}(x)$ is even, then $\operatorname{prog}(\nabla h_2(x)) \le \operatorname{prog}(x)$. Therefore, the workers, starting from

any point with any algorithm $A \in \mathcal{A}^U_{\{C_i\}_{i=1}^n}$, can only earn one more non-zero coordinate if they do not synchronize (communicate) with the server; after that, the number of non-zero coordinates of local models will not increase any more. That is to say, in order to proceed (i.e.,, achieve more non-zero coordinates), the worker must synchronize the gradient information, via the server, between the odd and even components.

(Step 2.) We consider a gradient oracle that can return the lossless full-batch gradient, i.e., $O_i(x) = \nabla f_i(x), \forall\, x \in \mathbb{R},\ 1 \le i \le n$. For the construction of ω-unbiased compressors, we consider $\{C_i\}_{i=1}^n$ to be the $\frac{d}{s}\times$rand-s operators with shared randomness and $s = \lceil d/(1+\omega)\rceil$, where the $\frac{d}{s}$-scaling procedure is to ensure unbiasedness. Specifically, during a round of communication, all workers randomly choose s coordinates from the full vector to be communicated, and then transmit the $\frac{d}{s}$-scaled values at chosen coordinates. The chosen coordinate indexes are identical across all workers due to the shared randomness and are sampled uniformly randomly per communication. Since for any $1 \le k \le d$, the index k has probability s/d to be chosen, we have for any $x \in \mathbb{R}^d$

$$\mathbb{E}[C_i(x)] = \mathbb{E}\left[\left(\frac{d}{s}x_k \mathbb{1}\{k \text{ is chosen}\}\right)_{1\le k\le d}\right] = \left(\frac{d}{s}x_i \mathbb{P}(k \text{ is chosen})\right)_{1\le k\le d} = x,$$

and

$$\begin{aligned}\mathbb{E}[\|C_i(x)-x\|^2] &= \sum_{k=1}^d \mathbb{E}\left[\left(\frac{d}{s}x_k\mathbb{1}\{k \text{ is chosen}\} - x_k\right)^2\right]\\ &= \sum_{k=1}^d x_k^2\left(\left(\frac{d}{s}-1\right)^2 \mathbb{P}(k \text{ is chosen}) + \mathbb{P}(k \text{ is not chosen})\right) = \sum_{k=1}^d x_k^2\left(\frac{d}{s}-1\right) \le \omega\|x\|^2\end{aligned}$$

where the last inequality follows the definition of s. Therefore, the above construction ensures $\{C_i\}_{i=1}^n \subseteq \mathcal{U}_\omega$.

For any $t = 0, 1, \ldots$ and $1 \le i \le n$, let $v_i^{(t,\star)}$ be the synchronized point that worker i wants to send to the server at the t-th communication. Due to communication compression, the server can only receive a compressed version $C_i(v_i^{(t,\star)})$ instead, which we denote by $v_i^{(t)} \triangleq C_i(v_i^{(t,\star)})$. It is easy to see that

$$\mathrm{prog}(v_i^{(t)}) \le \mathrm{prog}(v_i^{(t,\star)}), \quad \forall\, v_i^{(t,\star)} \in \mathbb{R}^d. \tag{15}$$

Furthermore, since the compressor C_i only takes s coordinates, $v_i^{(t)}$ has probability $1 - s/d \approx \frac{\omega}{1+\omega}$ in making the coordinate with index $\mathrm{prog}(v_i^{(t,\star)})$ zero, which implies $\mathbb{P}(\mathrm{prog}(v^{(t)}) < \mathrm{prog}(v^{(t,\star)})) \approx \frac{\omega}{1+\omega}$. This is to say, each worker i has only probability $\approx (1+\omega)^{-1}$ to transmit its last non-zero entry. Therefore, algorithms are hampered by the compressors to achieve more non-zero coordinates for model parameters in synchronizing the gradient information from all workers.

(Step 3.) Since we only consider algorithms satisfying the zero-respecting property, as discussed in [16, 17, 41], worker i, within T rounds of communication, can only return model $\hat{x}$ such that

$$\hat{x} \in \mathrm{span}\left(\left\{x^{(0)}, \nabla f_i(x^{(0)}), \left\{\{v_i^{(t)}, \nabla f_i(v_i^{(t)}) : 0 \le t < T\} : 1 \le i \le n\right\}\right\}\right). \tag{16}$$

Since f_is are zero-chain functions, $\hat{x}_i$ satisfies

$$\begin{aligned}\mathrm{prog}(\hat{x}) &\le \mathrm{prog}\left(\left\{x^{(0)}, \nabla f_i(x^{(0)}), \left\{\{v_i^{(t)}, \nabla f_i(v_i^{(t)}) : 0 \le t < T\} : 1 \le i \le n\right\}\right\}\right)\\ &\le \max_{0\le t<T}\max_{1\le i\le n} \mathrm{prog}(v_i^{(t)}) + 1.\end{aligned} \tag{17}$$

By Lemma 5, we have

$$\mathbb{P}(\max_{0\le t<T}\max_{1\le i\le n} \mathrm{prog}(v_i^{(t)}) \ge d-1) \le e^{(e-1)T\lceil d/(1+\omega)\rceil/d+1-d}. \tag{18}$$

Combining (18) with (17) and (12), we have that

$$\mathbb{E}[\|\nabla f(\hat{x})\|^2] \ge (1 - e^{(e-1)T\lceil d/(1+\omega)\rceil/d+1-d})\frac{L^2\lambda^2}{\ell_0^2}. \tag{19}$$

Let

$$\lambda = \frac{\ell_0}{L}\sqrt{\frac{(1+\omega)\Delta L}{5T\ell_0\delta_0}} \quad \text{and} \quad d = \lfloor 5T/(1+\omega)\rfloor. \tag{20}$$

Then (8) naturally holds. Since T is assumed to be no less than $(1+\omega)^2$, we have $d = \lfloor 5T/(1+\omega)\rfloor \geq 5T/(1+\omega) - 1 \geq 4T/(1+\omega) \geq 4(1+\omega) \geq 4$. Then it is easy to verify

$$\begin{aligned}
&(e-1)T\left\lceil \frac{d}{1+\omega}\right\rceil /d + 1 - d \leq (e-1)T\left(\frac{d}{1+\omega}+1\right)/d + 1 - d \\
=&(e-1)\frac{T}{1+\omega} + (e-1)T/d + 1 - d \leq (e-1)\frac{T}{1+\omega} + (e-1)\frac{T}{4(1+\omega)} + 1 - \frac{4T}{1+\omega} \\
=&((e-1)(1+\frac{1}{4}) - 4)\frac{T}{1+\omega} + 1 \leq \frac{5e-17}{4} < 0,
\end{aligned}$$

which, combined with (19), further implies

$$\mathbb{E}[\|\nabla f(\hat{x})\|^2] = \Omega\left(\frac{L^2\lambda^2}{\ell_0^2}\right) = \Omega\left(\frac{\Delta L}{5\delta T\ell_0\delta_0}\right) = \Omega\left(\frac{(1+\omega)\Delta L}{T}\right).$$

Lemma 5. *In Example 2 in the proof of Theorem 1, it holds that*

$$\mathbb{P}(\max_{0\leq t<T}\max_{1\leq i\leq n}\mathrm{prog}(v_i^{(t)}) \geq d-1) \leq e^{(e-1)T\lceil d/(1+\omega)\rceil/d-d}$$

for any $T \geq 0$.

Proof. Note that for any $0 \leq t < T$, at the $(t+1)$-th round of communication, worker i can only transmit vector $v_i^{(t,\star)}$ that is aggregated from local gradient descent and the vectors in the past communication. Therefore, $v_i^{(t,\star)}$ satisfies

$$\begin{aligned}
\mathrm{prog}(v_i^{(t,\star)}) &\leq \mathrm{prog}\left(\left\{x^{(0)}, \nabla f_i(x^{(0)}), \left\{\{v_i^{(s)}, \nabla f_i(v_i^{(s)}) : 0 \leq s < t\} : 1 \leq i \leq n\right\}\right\}\right) \\
&\leq \max_{0\leq s<t}\max_{1\leq i\leq n}\mathrm{prog}(v_i^{(s)}) + 1 \triangleq B^{(t)}.
\end{aligned} \tag{21}$$

We define $B^{(0)} = 1$ additionally. By the definition of $B^{(t)}$ and that $\mathrm{prog}(v^{(t,\star)}) \leq \mathrm{prog}(v_i^{(t,\star)})$, it naturally holds that

$$\begin{aligned}
B^{(t)} \leq B^{(t+1)} &= \max_{0\leq s<(t+1)}\max_{1\leq i\leq n}\mathrm{prog}(v_i^{(s)}) + 1 = \max\left\{B^{(t)}, \max_{1\leq i\leq n}\mathrm{prog}(v_i^{(t)})\right\} + 1 \\
&\leq \max\left\{B^{(t)}, \max_{1\leq i\leq n}\mathrm{prog}(v_i^{(t,\star)})\right\} + 1 \overset{(21)}{\leq} B^{(t)} + 1.
\end{aligned} \tag{22}$$

Therefore, one round of communication can increase $B^{(t)}$ at most by 1.

From (22), we see that $B^{(t+1)} = B^{(t)} + 1$ only when $\max_{1\leq i\leq n}\mathrm{prog}(v_i^{(t)}) = \max_{1\leq i\leq n}\mathrm{prog}(v_i^{(t,\star)})$. Let $k = \max_{1\leq i\leq n}\mathrm{prog}(v_i^{(t,\star)})$. Recall that the compressors constructed in Example 2 are built on the shared randomness, therefore $\max_{1\leq i\leq n}\mathrm{prog}(v_i^{(t)}) = \max_{1\leq i\leq n}\mathrm{prog}(v_i^{(t,\star)}) = k$ is equivalent to that the coordinate index k is chosen during the $(t+1)$-round of communication compression, which is of probability $\frac{s}{d}$. Therefore, we have

$$\begin{aligned}
&\mathbb{P}(B^{(t+1)} = B^{(t)} + 1) \leq \mathbb{P}(\max_{1\leq i\leq n}\mathrm{prog}(v_i^{(t)}) = \max_{1\leq i\leq n}\mathrm{prog}(v_i^{(t,\star)})) \\
=&\mathbb{P}\left(\text{the coordinate index } \max_{1\leq i\leq n}\mathrm{prog}(v_i^{(t,\star)}) \text{ is chosen}\right) = \frac{s}{d}.
\end{aligned} \tag{23}$$

Let $E^{(t+1)}$ be the event $\{$the coordinate index $\max_{1\leq i\leq n}\mathrm{prog}(v_i^{(t,\star)})$ is chosen during the $(t+1)$-round of compression$\}$. Since the compression is uniformly random, we have

$\mathbb{1}(E^{(1)}), \ldots, \mathbb{1}(E^{(T)}) \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(\frac{s}{d})$ where $\mathbb{1}(\cdot)$ is the indicator function. By the above argument, we also have $B^{(t+1)} - B^{(t)} \leq \mathbb{1}(E^{(t+1)})$ for any $0 \leq t < T$. Therefore, we have

$$\begin{aligned}
\mathbb{P}(B^{(T)} \geq d) &\leq e^{-d}\mathbb{E}[e^{B^{(T)}}] = e^{-d}\mathbb{E}\left[\exp\left(B^{(0)} + \sum_{t=0}^{T-1}(B^{(t+1)} - B^{(t)})\right)\right] \\
&\leq e^{-(d-1)}\mathbb{E}\left[\exp\left(\sum_{t=0}^{T-1}\mathbb{1}(E^{(t+1)})\right)\right] = e^{-(d-1)}\prod_{t=0}^{T-1}\mathbb{E}\left[\exp\left(\mathbb{1}(E^{(t+1)})\right)\right] \\
&= e^{-(d-1)}\prod_{t=0}^{T-1}\left(1 + \frac{s}{d}(e-1)\right) \leq e^{-(d-1)}\prod_{t=0}^{T-1} e^{(e-1)s/d} = e^{(e-1)Ts/d-d+1},
\end{aligned}$$

which directly leads to the conclusion. □

A.2 Proof of Theorem 2

Theorem 2 essentially follows the same analysis as in Theorem 1. The only difference is that we shall construct compressors in proving $\Omega(\frac{\Delta L}{\delta T})$ by using rand-s operators with shared randomness and $s = \lceil \delta d \rceil$. There is no scaling procedure in compression. We can easily check that

$$\begin{aligned}
\mathbb{E}[\|C_i(x) - x\|^2] &= \sum_{k=1}^{d}\mathbb{E}\left[(x_k\mathbb{1}\{k \text{ is chosen}\} - x_k)^2\right] \\
&= \sum_{k=1}^{d} x_k^2\mathbb{P}(k \text{ is not chosen}) = \left(1 - \frac{s}{d}\right)\|x\|^2 \leq \delta\|x\|^2.
\end{aligned}$$

where the last inequality follows the definition of s. Therefore, we have $\{C_i\}_{i=1}^n \subseteq \mathcal{C}_\delta$. The scaling procedure does not change prog, and thus makes no effect on the argument in terms of non-zero coordinates. By considering $\omega = \delta^{-1} - 1$, i.e., $1 + \omega = \delta$, we can easily adapt the proof of Theorem 1 to reach Theorem 2.

B Convergence of NEOLITHIC

B.1 Proof of Lemma 2

Let $v^{(k,r)}$ be the r-th intermediate point generated in FCC for any $0 \leq r \leq R$. Since C is δ-contractive, we have

$$\begin{aligned}
\mathbb{E}[\|v^{(k,R)} - v^{(k,\star)}\|^2] &= \mathbb{E}[\|v^{(k,R-1)} + C(v^{(k,\star)} - v^{(k,R-1)}) - v^{(k,\star)}\|^2] \\
&= \mathbb{E}\left[\mathbb{E}[\|C(v^{(k,\star)} - v^{(k,R-1)}) + v^{(k,R-1)} - v^{(k,\star)}\|^2 \mid v^{(k,R-1)}]\right] \\
&\leq (1-\delta)\mathbb{E}[\|v^{(k,R-1)} - v^{(k,\star)}\|^2] \\
&\leq (1-\delta)^2\mathbb{E}[\|v^{(k,R-2)} - v^{(k,\star)}\|^2] \\
&\leq (1-\delta)^R\mathbb{E}[\|v^{(k,0)} - v^{(k,\star)}\|^2] = (1-\delta)^R\|v^{(k,\star)}\|^2.
\end{aligned}$$

B.2 Proof of Theorem 3

In this subsection, we provide the convergence proof for NEOLITHIC with bidirectional compression and contractive compressors. We first introduce some notations: $\Omega^{(k)} := \delta^{(k)} + \frac{1}{n}\sum_{i=1}^n \delta_i^{(k)}$, $\forall\, k \geq -1$; $y^{(k)} := x^{(k)} - \gamma\Omega^{(k-1)}$, $\forall\, k \geq 0$; $\Psi^{(k)} := \|\delta^{(k)}\|^2 + \frac{3}{n}\sum_{i=1}^n \|\delta_i^{(k)}\|^2$, $\forall\, k \geq -1$. We will use the following lemmas.

Lemma 6 (RECURSION FORMULA)**.** *For any $k \geq 0$, it holds that*

$$x^{(k+1)} - x^{(k)} = -\frac{\gamma}{n}\sum_{i=1}^n \hat{g}_i^{(k)} - \gamma\Omega^{(k-1)} + \gamma\Omega^{(k)}.$$

Proof. It is observed from Algorithm 1 that the vector $v^{(k,R)}$ returned by the FCC operator satisfies

$$v^{(k,R)} = \sum_{r=0}^{R-1} c^{(k,r)} \quad (\text{or } v_i^{(k,R)} = \sum_{r=0}^{R-1} c_i^{(k,r)} \text{ if FCC is utilized in node } i) \tag{24}$$

With (24), the relation between $\tilde{g}^{(k)}$ and $\delta^{(k)}$ (or between $\tilde{g}_i^{(k)}$ and $\delta_i^{(k)}$) in Algorithm 2 satisfies

$$\tilde{g}^{(k)} - \delta^{(k)} = \sum_{r=0}^{R-1} c^{(k,r)} \quad \text{and} \quad \tilde{g}_i^{(k)} - \delta_i^{(k)} = \sum_{r=0}^{R-1} c_i^{(k,r)}, \quad \forall\, 1 \le i \le n. \tag{25}$$

Therefore, we have for any $k \ge 0$

$$\begin{aligned}
x^{(k+1)} - x^{(k)} &= -\gamma \sum_{r=0}^{R-1} c^{(k,r)} \overset{(25)}{=} -\gamma(\tilde{g}^{(k)} - \delta^{(k)}) \\
&= -\gamma \left(\delta^{(k-1)} + \frac{1}{n} \sum_{i=1}^{n} \sum_{r=0}^{R-1} c_i^{(k,r)} - \delta^{(k)} \right) \qquad (26) \\
&\overset{(25)}{=} -\gamma \left(\delta^{(k-1)} + \frac{1}{n} \sum_{i=1}^{n} \left(\tilde{g}_i^{(k)} - \delta_i^{(k)} \right) - \delta^{(k)} \right) \\
&= -\gamma \left(\delta^{(k-1)} + \frac{1}{n} \sum_{i=1}^{n} \left(\hat{g}_i^{(k)} + \delta_i^{(k-1)} - \delta_i^{(k)} \right) - \delta^{(k)} \right) \qquad (27) \\
&= -\frac{\gamma}{n} \sum_{i=1}^{n} \hat{g}_i^{(k)} - \gamma \Omega^{(k-1)} + \gamma \Omega^{(k)} \qquad (28)
\end{aligned}$$

where (26) and (27) follow the implementation of Algorithm 2, and we use the notation of $\Omega^{(k)}$ in (28). □

Lemma 7 (DESCENT LEMMA)**.** *Let the auxiliary sequence be $y^{(k)} := x^{(k)} - \gamma \Omega^{(k-1)}$, $\forall\, k \ge 0$. Under Assumption 1, if learning rate $0 < \gamma \le \frac{1}{2L}$, it holds that for any $k \ge 0$,*

$$\mathbb{E}[f(y^{(k+1)})] \le \mathbb{E}[f(y^{(k)})] - \frac{\gamma}{4} \mathbb{E}[\|\nabla f(x^{(k)})\|^2] + 2\gamma^3 L^2 \mathbb{E}[\|\Omega^{(k-1)}\|^2] + \frac{\gamma^2 L \sigma^2}{2nR}. \tag{29}$$

Proof. By Lemma 6 and the definition of $y^{(k)}$ for $k \ge 0$, we directly have that

$$y^{(k+1)} = y^{(k)} - \frac{\gamma}{n} \sum_{i=1}^{n} \hat{g}_i^{(k)}.$$

Since f is L-smooth, we have

$$\begin{aligned}
f(y^{(k+1)}) &\le f(y^{(k)}) + \langle \nabla f(y^{(k)}), y^{(k+1)} - y^{(k)} \rangle + \frac{L}{2} \|y^{(k+1)} - y^{(k)}\|^2 \\
&= f(y^{(k)}) - \gamma \left\langle \nabla f(y^{(k)}), \frac{1}{n} \sum_{i=1}^{n} \hat{g}_i^{(k)} \right\rangle + \frac{\gamma^2 L}{2} \left\| \frac{1}{n} \sum_{i=1}^{n} \hat{g}_i^{(k)} \right\|^2. \qquad (30)
\end{aligned}$$

Since $\hat{g}_i^{(k)} = \frac{1}{R} \sum_{r=1}^{R} O_i(x^{(k)}; \zeta_i^{(k,r)})$ is a unbiased estimator of $\nabla f_i(x^{(k)})$, and by Assumption 2, we have

$$\mathbb{E}\left[\frac{1}{n} \sum_{i=1}^{n} \hat{g}_i^{(k)} \right] = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(x^{(k)}) = \nabla f(x^{(k)}) \tag{31}$$

and

$$\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n}\hat{g}_i^{(k)}\right\|^2\right] = \|\nabla f(x^{(k)})\|^2 + \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n}\hat{g}_i^{(k)} - \nabla f(x^{(k)})\right\|^2\right]$$
$$= \|\nabla f(x^{(k)})\|^2 + \mathbb{E}\left[\left\|\frac{1}{nR}\sum_{i=1}^{n}\sum_{r=0}^{R-1}\left(O_i(x^{(k)};\zeta_i^{(k,r)}) - \nabla f_i(x^{(k)})\right)\right\|^2\right]$$
$$\leq \|\nabla f(x^{(k)})\|^2 + \frac{\sigma^2}{nR}. \tag{32}$$

Taking global expectation over (30), and plugging (31) and (32) into it, we reach

$$\begin{aligned}
&\mathbb{E}[f(y^{(k+1)})] - \mathbb{E}[f(y^{(k)})] \\
\leq& -\gamma\mathbb{E}[\langle\nabla f(y^{(k)}), \nabla f(x^{(k)})\rangle] + \frac{\gamma^2 L}{2}\mathbb{E}[\|\nabla f(x^{(k)})\|^2] + \frac{\gamma^2 L\sigma^2}{2nR} \\
=& -\gamma\mathbb{E}[\langle\nabla f(y^{(k)}) - \nabla f(x^{(k)}), \nabla f(x^{(k)})\rangle] - \left(\gamma - \frac{\gamma^2 L}{2}\right)\mathbb{E}[\|\nabla f(x^{(k)})\|^2] + \frac{\gamma^2 L\sigma^2}{2nR} \\
\leq& 2\gamma\mathbb{E}[\|\nabla f(y^{(k)}) - \nabla f(x^{(k)})\|^2] + \frac{\gamma}{2}\mathbb{E}[\|\nabla f(x^{(k)})\|^2] \\
&- \left(\gamma - \frac{\gamma^2 L}{2}\right)\mathbb{E}[\|\nabla f(x^{(k)})\|^2] + \frac{\gamma^2 L\sigma^2}{2nR} && (33) \\
\leq& 2\gamma L^2\mathbb{E}[\|y^{(k)} - x^{(k)}\|^2] - \frac{\gamma(1-\gamma L)}{2}\mathbb{E}[\|\nabla f(x^{(k)})\|^2] + \frac{\gamma^2 L\sigma^2}{2nR}, && (34)
\end{aligned}$$

where we use Young's inequality in (33), and (34) holds by Assumption 1. Using $y^{(k)} - x^{(k)} = -\gamma\Omega^{(k-1)}$ and that

$$0 < \gamma \leq \frac{1}{2L} \quad \Longrightarrow \quad \frac{\gamma(1-\gamma L)}{2} \geq \frac{\gamma}{4}$$

in (34), we reach the conclusion in this lemma. □

Lemma 8 (VANISHING ERROR). *Assume R is sufficiently large such that $(1-\delta)^R < \frac{1}{4}$ and let $\Psi^{(k)} := \|\delta^{(k)}\|^2 + \frac{3}{n}\sum_{i=1}^{n}\|\delta_i^{(k)}\|^2$ for $k \geq -1$. Under Assumptions 2, 4, 5 It holds that*

$$\mathbb{E}[\Psi^{(k)}] \leq 4(1-\delta)^R\left(\mathbb{E}[\Psi^{(k-1)}] + 5\mathbb{E}[\|\nabla f(x^{(k)})\|^2] + 4\left(b^2 + \frac{\sigma^2}{R}\right)\right).$$

Proof. By Lemma 2, it holds that

$$\mathbb{E}[\|\delta^{(k)}\|^2] = \mathbb{E}[\|\tilde{g}^{(k)} - \text{FCC}(\tilde{g}^{(k)}, C, R)\|^2] \leq (1-\delta)^R\mathbb{E}[\|\tilde{g}^{(k)}\|^2].$$

Note that

$$\begin{aligned}
\tilde{g}^{(k)} &= \delta^{(k-1)} + \frac{1}{n}\sum_{i=1}^{n}\sum_{r=0}^{R-1}c_i^{(k,r)} = \delta^{(k-1)} + \frac{1}{n}\sum_{i=1}^{n}\left(\tilde{g}_i^{(k)} - \delta_i^{(k)}\right) \\
&= \delta^{(k-1)} + \frac{1}{n}\sum_{i=1}^{n}\left(\hat{g}_i^{(k)} + \delta_i^{(k-1)} - \delta_i^{(k)}\right) \\
&= \delta^{(k-1)} + \frac{1}{n}\sum_{i=1}^{n}\hat{g}_i^{(k)} + \frac{1}{n}\sum_{i=1}^{n}\delta_i^{(k-1)} - \frac{1}{n}\sum_{i=1}^{n}\delta_i^{(k)}.
\end{aligned}$$

Therefore, by using Young's inequality and (32), we have

$$\begin{aligned}
\mathbb{E}[\|\delta^{(k)}\|^2] &\leq (1-\delta)^R \mathbb{E}[\|\tilde{g}^{(k)}\|^2] \\
&\leq (1-\delta)^R \Bigg(4\mathbb{E}[\|\delta^{(k-1)}\|^2] + 4\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \hat{g}_i^{(k)}\right\|^2\right] \\
&\qquad + \frac{4}{n}\sum_{i=1}^n \mathbb{E}[\|\delta_i^{(k)}\|^2] + \frac{4}{n}\sum_{i=1}^n \mathbb{E}[\|\delta_i^{(k-1)}\|^2] \Bigg) \\
&\leq (1-\delta)^R \Bigg(4\mathbb{E}[\|\delta^{(k-1)}\|^2] + 4\mathbb{E}[\|\nabla f(x^{(k)})\|^2] \\
&\qquad + \frac{4}{n}\sum_{i=1}^n \mathbb{E}[\|\delta_i^{(k)}\|^2] + \frac{4}{n}\sum_{i=1}^n \mathbb{E}[\|\delta_i^{(k-1)}\|^2] + \frac{4\sigma^2}{nR} \Bigg). \qquad (35)
\end{aligned}$$

For any $1 \leq i \leq n$, using the similar argument, we have that

$$\begin{aligned}
\mathbb{E}[\|\delta_i^{(k)}\|^2] =& \mathbb{E}[\|\tilde{g}^{(k)} - \text{FCC}(\tilde{g}_i^{(k)}, C_i, R)\|^2] \leq (1-\delta)^R \mathbb{E}[\|\tilde{g}_i^{(k)}\|^2] \\
=& (1-\delta)^R \mathbb{E}[\|\hat{g}_i^{(k)} + \delta_i^{(k-1)}\|^2] \\
\leq& (1-\delta)^R \left(2\mathbb{E}[\|\hat{g}_i^{(k)}\|^2] + 2\mathbb{E}[\|\delta_i^{(k-1)}\|^2] \right) \\
=& (1-\delta)^R \left(2\mathbb{E}[\|\nabla f_i(x^{(k)})\|^2] + 2\mathbb{E}[\|\delta_i^{(k-1)}\|^2] + \frac{2\sigma^2}{R} \right). \qquad (36)
\end{aligned}$$

Note that for any $1 \leq i \leq n$,

$$\begin{aligned}
\mathbb{E}[\|\nabla f_i(x^{(k)})\|^2] &= \mathbb{E}[\|\nabla f_i(x^{(k)}) - \nabla f(x^{(k)}) + \nabla f(x^{(k)})\|^2] \\
&\leq 2\mathbb{E}[\|\nabla f_i(x^{(k)}) - \nabla f(x^{(k)})\|^2] + 2\mathbb{E}[\|\nabla f(x^{(k)})\|^2]. \qquad (37)
\end{aligned}$$

Taking the average of (36) over all $1 \leq i \leq n$, and using (37) and Assumption 5, we further have that

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^n \mathbb{E}[\|\delta_i^{(k)}\|^2] \\
=& (1-\delta)^R \Bigg(\frac{4}{n}\sum_{i=1}^n \mathbb{E}[\|\nabla f_i(x^{(k)}) - \nabla f(x^{(k)})\|^2] + 4\mathbb{E}[\|\nabla f(x^{(k)})\|^2] \\
&\qquad + \frac{2}{n}\sum_{i=1}^n \mathbb{E}[\|\delta_i^{(k-1)}\|^2] + \frac{2\sigma^2}{R} \Bigg) \\
\leq& (1-\delta)^R \left(4\mathbb{E}[\|\nabla f(x^{(k)})\|^2] + \frac{2}{n}\sum_{i=1}^n \mathbb{E}[\|\delta_i^{(k-1)}\|^2] + 4b^2 + \frac{2\sigma^2}{R} \right). \qquad (38)
\end{aligned}$$

Summing up (35) and plugging (38) into it, we obtain

$$\begin{aligned}
&\mathbb{E}[\|\delta^{(k)}\|^2] + \frac{4}{n}\sum_{i=1}^n \mathbb{E}[\|\delta_i^{(k)}\|^2] \\
\leq& (1-\delta)^R \Bigg(4\mathbb{E}[\|\delta^{(k-1)}\|^2] + 4\mathbb{E}[\|\nabla f(x^{(k)})\|^2] + \frac{4}{n}\sum_{i=1}^n \mathbb{E}[\|\delta_i^{(k)}\|^2] + \frac{4}{n}\sum_{i=1}^n \mathbb{E}[\|\delta_i^{(k-1)}\|^2] \\
&\qquad + \frac{4\sigma^2}{nR} + 16\mathbb{E}[\|\nabla f(x^{(k)})\|^2] + \frac{8}{n}\sum_{i=1}^n \mathbb{E}[\|\delta_i^{(k-1)}\|^2] + 16b^2 + \frac{8\sigma^2}{R} \Bigg) \\
\leq& (1-\delta)^R \Bigg(4\mathbb{E}[\|\delta^{(k-1)}\|^2] + \frac{4}{n}\sum_{i=1}^n \mathbb{E}[\|\delta_i^{(k)}\|^2] + \frac{12}{n}\sum_{i=1}^n \mathbb{E}[\|\delta_i^{(k-1)}\|^2] \\
&\qquad + 20\mathbb{E}[\|\nabla f(x^{(k)})\|^2] + 16b^2 + \frac{12\sigma^2}{R} \Bigg).
\end{aligned}$$

We thus have

$$\mathbb{E}[\|\delta^{(k)}\|^2] + \frac{4(1-(1-\delta)^R)}{n}\sum_{i=1}^n \mathbb{E}[\|\delta_i^{(k)}\|^2]$$
$$\leq (1-\delta)^R\left(4\mathbb{E}[\|\delta^{(k-1)}\|^2] + \frac{12}{n}\sum_{i=1}^n \mathbb{E}[\|\delta_i^{(k-1)}\|^2] + 20\mathbb{E}[\|\nabla f(x^{(k)})\|^2] + 16b^2 + \frac{12\sigma^2}{R}\right).$$

Since R is sufficiently large such that

$$(1-\delta)^R < \frac{1}{4}, \tag{39}$$

we have that $4(1-(1-\delta)^R) \geq 3$ and hence

$$\mathbb{E}[\Psi^{(k)}] \leq \mathbb{E}[\|\delta^{(k)}\|^2] + \frac{4(1-(1-\delta)^R)}{n}\sum_{i=1}^n \mathbb{E}[\|\delta_i^{(k)}\|^2]$$
$$\leq 4(1-\delta)^R\left(\mathbb{E}[\Psi^{(k-1)}] + 5\mathbb{E}[\|\nabla f(x^{(k)})\|^2] + 4\left(b^2 + \frac{\sigma^2}{R}\right)\right).$$

□

With Lemma 8, we easily reach its ergodic version:

Lemma 9 (ERGODIC VANISHING ERROR)**.** *Suppose $\theta \triangleq 4(1-\delta)^R < 1$. Then it holds that for any $K \geq 0$,*

$$\frac{1}{K+1}\sum_{k=0}^K \mathbb{E}[\Psi^{(k-1)}] \leq \frac{1}{K+1}\frac{5\theta}{1-\theta}\sum_{\ell=0}^{K-1}\mathbb{E}[\|\nabla f(x^{(\ell)})\|^2] + \frac{4\theta}{1-\theta}\left(b^2 + \frac{\sigma^2}{R}\right).$$

Proof. Let $\theta = 4(1-\delta)^R$, then by Lemma 8 and noting $\Psi^{(-1)} = 0$, we have

$$\begin{aligned}
\mathbb{E}[\Psi^{(k)}] \leq& \theta\left(\mathbb{E}[\Psi^{(k-1)}] + 5\mathbb{E}[\|\nabla f(x^{(k)})\|^2] + 4\left(b^2 + \frac{\sigma^2}{R}\right)\right)\\
\leq& \theta\Bigg(\theta\left(\mathbb{E}[\Psi^{(k-2)}] + 5\mathbb{E}[\|\nabla f(x^{(k-1)})\|^2] + 4\left(b^2 + \frac{\sigma^2}{R}\right)\right)\\
&+ 5\mathbb{E}[\|\nabla f(x^{(k)})\|^2] + 4\left(b^2 + \frac{\sigma^2}{R}\right)\Bigg)\\
=& \theta^2\mathbb{E}[\Psi^{(k-2)}] + 5\sum_{\ell=k-1}^k \theta^{k+1-\ell}\mathbb{E}[\|\nabla f(x^{(\ell)})\|^2] + 4\sum_{\ell=k-1}^k \theta^{k+1-\ell}\left(b^2 + \frac{\sigma^2}{R}\right)\\
\leq& \cdots\\
\leq& \theta^{k+1}\mathbb{E}[\Psi^{(-1)}] + 5\sum_{\ell=0}^k \theta^{k+1-\ell}\mathbb{E}[\|\nabla f(x^{(\ell)})\|^2] + 4\sum_{\ell=0}^k \theta^{k+1-\ell}\left(b^2 + \frac{\sigma^2}{R}\right)\\
=& 5\sum_{\ell=0}^k \theta^{k+1-\ell}\mathbb{E}[\|\nabla f(x^{(\ell)})\|^2] + 4\sum_{\ell=0}^k \theta^{k+1-\ell}\left(b^2 + \frac{\sigma^2}{R}\right).
\end{aligned} \tag{40}$$

Therefore, by taking the summation of (40) over $k = 0, \ldots, K-1$ and using $\Psi^{(-1)} = 0$ again, we further have

$$\frac{1}{K+1}\sum_{k=0}^K \mathbb{E}[\Psi^{(k-1)}] = \frac{1}{K+1}\sum_{k=0}^{K-1}\mathbb{E}[\Psi^{(k)}]$$
$$\leq \frac{1}{K+1}\sum_{k=0}^{K-1}\left(5\sum_{\ell=0}^k \theta^{k+1-\ell}\mathbb{E}[\|\nabla f(x^{(\ell)})\|^2] + 4\sum_{\ell=0}^k \theta^{k+1-\ell}\left(b^2 + \frac{\sigma^2}{R}\right)\right)$$
$$= \frac{1}{K+1}\left(5\sum_{\ell=0}^{K-1}\mathbb{E}[\|\nabla f(x^{(\ell)})\|^2]\left(\sum_{k=\ell}^{K-1}\theta^{k+1-\ell}\right) + 4\left(b^2 + \frac{\sigma^2}{R}\right)\sum_{\ell=0}^{K-1}\sum_{k=\ell}^{K-1}\theta^{k+1-\ell}\right),$$

where we change the summation order of indexes k and ℓ in the last identity. Since $\sum_{k=\ell}^{K-1}\theta^{k+1-\ell} = \frac{\theta(1-\theta^{K-\ell})}{1-\theta} \leq \frac{\theta}{1-\theta}$, we thus have

$$\begin{aligned}
&\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}[\Psi^{(k-1)}] \\
\leq&\frac{1}{K+1}\left(5\sum_{\ell=0}^{K-1}\mathbb{E}[\|\nabla f(x^{(\ell)})\|^2]\frac{\theta}{1-\theta}+4\left(b^2+\frac{\sigma^2}{R}\right)\frac{K\theta}{1-\theta}\right) \\
\leq&\frac{1}{K+1}\frac{5\theta}{1-\theta}\sum_{\ell=0}^{K-1}\mathbb{E}[\|\nabla f(x^{(\ell)})\|^2]+\frac{4\theta}{1-\theta}\left(b^2+\frac{\sigma^2}{R}\right).
\end{aligned}$$

□

Given the above lemmas, now we prove the convergence rate of NEOLITHIC.

Theorem 4. *Let the communication round be* $R = \left\lceil \frac{\ln(n/\delta)+\ln(4\max\{b^2,\delta\sigma^2\})}{\delta}\right\rceil$ *and learning rate be as in* (46)*. Then it holds that for any* $K \geq 0$*,*

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

where $T = KR$ *is the total number of gradient queries (compressed communications) on each worker.*

Proof. Averaging (29) over $k = 0, \ldots, K$, and using the fact that $y^{(0)} = x^{(0)}$ and $f(y^{(K+1)}) \geq f^\star$, we have

$$\begin{aligned}
&\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}[\|\nabla f(x^{(k)})\|^2] \\
\leq&\frac{4\mathbb{E}[f(y^{(0)})]-4\mathbb{E}[f(y^{(K+1)})]}{\gamma(K+1)}+\frac{8\gamma^2L^2}{K+1}\sum_{k=0}^{K}\mathbb{E}[\|\Omega^{(k-1)}\|^2]+\frac{2\gamma L\sigma^2}{nR} \\
\leq&\frac{4(f(x^{(0)})-f^\star)}{\gamma(K+1)}+\frac{8\gamma^2L^2}{K+1}\sum_{k=0}^{K}\mathbb{E}[\|\Omega^{(k-1)}\|^2]+\frac{2\gamma L\sigma^2}{nR}. \qquad (41)
\end{aligned}$$

By the definition of $\Omega^{(k-1)}$ and using the Cauhy-Schwarz inequality, it holds that

$$\begin{aligned}
\mathbb{E}[\|\Omega^{(k-1)}\|^2] =&\mathbb{E}\left[\left\|\delta^{(k-1)}+\frac{1}{n}\sum_{i=1}^{n}\delta_i^{(k-1)}\right\|^2\right] \\
\leq&\left(1+\frac{1}{3}\right)\mathbb{E}[\|\delta^{(k-1)}\|^2]+(1+3)\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n}\delta_i^{(k-1)}\right\|^2\right] \\
\leq&\frac{4}{3}\mathbb{E}[\|\delta^{(k-1)}\|^2]+\frac{4}{n}\sum_{i=1}^{n}\mathbb{E}[\|\delta_i^{(k-1)}\|^2]=\frac{4}{3}\mathbb{E}[\Psi^{(k-1)}].
\end{aligned}$$

Therefore, by Lemma 9, we reach

$$\begin{aligned}
&\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}[\|\Omega^{(k-1)}\|^2] \leq \frac{4}{3(K+1)}\sum_{k=0}^{K}\mathbb{E}[\|\Psi^{(k-1)}\|^2] \\
\leq&\frac{20\theta}{3(K+1)(1-\theta)}\sum_{\ell=0}^{K-1}\mathbb{E}[\|\nabla f(x^{(\ell)})\|^2]+\frac{16\theta}{3(1-\theta)}\left(b^2+\frac{\sigma^2}{R}\right). \qquad (42)
\end{aligned}$$

Plugging (42) into (41), we reach that

$$\begin{aligned}
&\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}[\|\nabla f(x^{(k)})\|^2] \\
\leq&\frac{4(f(x^{(0)})-f^\star)}{\gamma(K+1)}+\frac{160\gamma^2L^2\theta}{3(1-\theta)(K+1)}\sum_{\ell=0}^{K-1}\mathbb{E}[\|\nabla f(x^{(\ell)})\|^2] \\
&+\frac{128\gamma^2L^2\theta}{3(1-\theta)}\left(b^2+\frac{\sigma^2}{R}\right)+\frac{2\gamma L\sigma^2}{nR}. \qquad (43)
\end{aligned}$$

Assume the learning rate γ is sufficiently small such that

$$\gamma\leq\frac{1}{11L}\sqrt{\frac{1-\theta}{\theta}} \quad\Longrightarrow\quad \frac{160\gamma^2L^2\theta}{3(1-\theta)}\leq\frac{1}{2}. \qquad (44)$$

then we can further bound (43) as

$$\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}[\|\nabla f(x^{(k)})\|^2]\leq\frac{8\Delta}{\gamma(K+1)}+\frac{4\gamma L\sigma^2}{nR}+\frac{256\gamma^2L^2\theta}{3(1-\theta)}\left(b^2+\frac{\sigma^2}{R}\right). \qquad (45)$$

where we use the definition $\Delta := f(x^{(0)})-f^\star$. By choosing

$$\gamma=\frac{1}{11L+\sigma\left(\frac{(K+1)L}{2nR\Delta}\right)^{\frac{1}{2}}+\left(\frac{32(K+1)L^2\theta(b^2+\sigma^2/R)}{3(1-\theta)\Delta}\right)^{\frac{1}{3}}}, \qquad (46)$$

we have

$$\begin{aligned}
&\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}[\|\nabla f(x^{(k)})\|^2] \\
\leq&\frac{16\sigma\sqrt{L\Delta}}{\sqrt{nR(K+1)}}+\frac{36\Delta^{\frac{2}{3}}L^{\frac{2}{3}}\theta^{\frac{1}{3}}(b^2+\sigma^2/R)^{\frac{1}{3}}}{(K+1)^{\frac{2}{3}}(1-\theta)^{\frac{1}{3}}}+\frac{88L\Delta}{K+1} \\
=&O\Big(\Big(\frac{\Delta L\sigma^2}{nT}\Big)^{\frac{1}{2}}+\frac{\theta^{\frac{1}{3}}\Delta^{\frac{2}{3}}L^{\frac{2}{3}}\max\{b^{\frac{2}{3}},\sigma^{\frac{2}{3}}/R^{\frac{1}{3}}\}}{(1-\theta)^{\frac{1}{3}}K^{\frac{2}{3}}}+\frac{\Delta L}{K}\Big) \qquad (47)
\end{aligned}$$

where we bound $f(x^{(0)})-f^\star$ by Δ in the last equation.

Let

$$R=\left\lceil\frac{\max\{\ln\left(\frac{\delta T\max\{b^2,\sigma^2\delta\}}{\Delta L}\right),\ln(8)\}}{\delta}\right\rceil=\tilde{O}\left(\frac{1}{\delta}\right), \qquad (48)$$

then it holds that

$$\begin{aligned}
\theta=&4(1-\delta)^R\leq 4e^{-\delta R} \\
\leq&4\exp\left(-\max\{\ln(\delta T\max\{b^2,\sigma^2\delta\}/\Delta L),\ln(8)\}\right)=\min\left\{\frac{4\Delta L}{\delta T\max\{b^2,\sigma^2\delta\}},\frac{1}{2}\right\}, \qquad (49)
\end{aligned}$$

and hence $1-\theta=\Omega(1)$ and γ satisfies (44). Plugging (48) and (49) into (47), and applying the notation $T=KR$, we reach

$$\begin{aligned}
\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}[\|\nabla f(x^{(k)})\|^2]=&O\Big(\Big(\frac{\Delta L\sigma^2}{nT}\Big)^{\frac{1}{2}}+\frac{\theta^{\frac{1}{3}}R^{\frac{2}{3}}\Delta^{\frac{2}{3}}L^{\frac{2}{3}}\max\{b^{\frac{2}{3}},\sigma^{\frac{2}{3}}/R^{\frac{1}{3}}\}}{T^{\frac{2}{3}}}+\frac{R\Delta L}{T}\Big) \\
=&\tilde{O}\Big(\Big(\frac{\Delta L\sigma^2}{nT}\Big)^{\frac{1}{2}}+\frac{\theta^{\frac{1}{3}}\Delta^{\frac{2}{3}}L^{\frac{2}{3}}\max\{b^{\frac{2}{3}},\sigma^{\frac{2}{3}}\delta^{\frac{1}{3}}\}}{\delta^{\frac{2}{3}}T^{\frac{2}{3}}}+\frac{\Delta L}{\delta T}\Big) \\
=&\tilde{O}\Big(\Big(\frac{\Delta L\sigma^2}{nT}\Big)^{\frac{1}{2}}+\frac{\Delta L}{\delta T}\Big). \qquad (50)
\end{aligned}$$

□

C Experiment Supplement

C.1 Synthetic Dataset

Linear regression. We consider the following least-square problem:

$$\min_{x \in \mathbb{R}^d} \quad \frac{1}{2n} \sum_{i=1}^{n} \|A_i x - b_i\|^2.$$

Coefficient matrix A_i and measurement b_i are associated with node i, and M is the size of local data. We set $d = 30$, $n = 32$ and $M = 1000$, and generate data by letting each node i be associated with a local solution $x_i^\star$ randomly generated by $\mathcal{N}(0, I_d)$. Then we generate each element in A_i following standard normal distribution, and measurement b_i is generated by $b_i = A_i x_i^\star + s_i$ with white noise $s_i \sim \mathcal{N}(0, 0.01)$. At each query, every node will randomly sample a row in A_i and the corresponding element in b_i to evaluate the stochastic gradient. We adopt the rand-1 compressor and set the number of rounds $R = 4$ for NEOLITHIC. We use stair-wise decaying learning rates in which the learning rates are divided by every $2,500$ communication rounds. Each algorithm is averaged with 20 trials. The result is shown in Figure 1 (left). It is observed that NEOLITHIC outperforms MEM-SGD and Double-Squeeze in convergence rate, and it performs closely to P-SGD.

Logistic regression. We consider the following logistic regression problem:

$$\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(x) \quad \text{where} \quad f_i(x) = \frac{1}{M} \sum_{m=1}^{M} \ln\left(1 + \exp\left(-y_{i,m} h_{i,m}^\top x\right)\right)$$

where $\{h_{i,m}, y_{i,m}\}_{m=1}^{M}$ is the training dateset held by node i in which $h_{i,m} \in \mathbb{R}^d$ with $d = 30$ is a feature vector while $y_{i,m} \in \{-1, +1\}$ is the corresponding label. Similar to the least square, each node i is associated with a local solution $x_i^\star$. We generate each feature vector $h_{i,m} \sim \mathcal{N}(0, I_d)$, and label $y_{i,m} = 1$ with probability $1/(1 + \exp(-y_{i,m} h_{i,m}^\top x_i^\star))$; otherwise $y_{i,m} = -1$. We adopt the rand-1 compressor and set the number of rounds $R = 4$ for NEOLITHIC. We use stair-wise decaying learning rates in which the learning rates are divided by every 800 communication rounds. Each algorithm is averaged with 20 trials. The result is shown in Figure 2. It is observed that NEOLITHIC outperforms MEM-SGD and Double-Squeeze in convergence rate, and it performs closely to P-SGD.

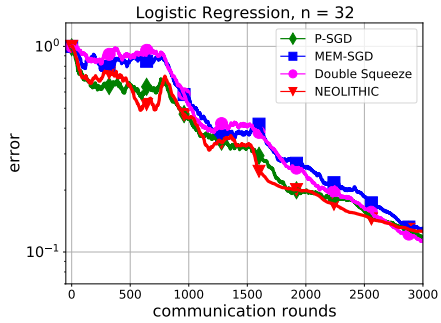

Figure 2: Convergence results on the synthetic logistic regression problem in terms of the mean-square error $\mathbb{E}[\|\nabla f(x)\|^2]$ versus communication rounds.

C.2 Deep Learning Tasks

Implementation details. We implement all compression algorithms with PyTorch[45] 1.8.2 using NCCL 2.8.3 (CUDA 10.1) as the communication backend. For PSGD, we used PyTorch's native Distributed Data Parallel (DDP) module. All deep learning training scripts in this section run on a server with 8 NVIDIA V100 GPUs in our cluster and each GPU is treated as one worker.

Image classification. We investigate the performance of the aforementioned methods with CIFAR-10 [35] dataset. For CIFAR-10 dataset, it consists of 50,000 training images and 10,000 validation images categorized in 10 classes. We utilize two common variants of ResNet [26] model on CIFAR-10 (ResNet-20 with roughly 0.27M parameters and ResNet-18 with 11.17M parameters). We train

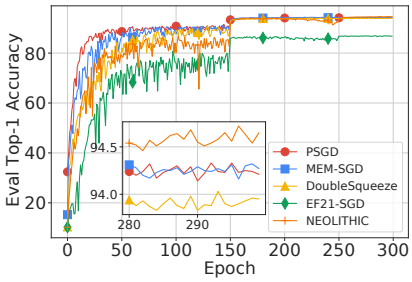

Figure 3: Convergence results on the CIFAR-10 in terms of validation accuracy.

Table 4: Accuracy comparison on CIFAR-10 with heterogeneous data (ResNet-20).

METHODS	MEM-SGD	DOUBLE-SQUEEZE	EF21-SGD	NEOLITHIC	C-RATIO %
$\alpha = 1$	85.42 ± 0.22	83.72 ± 0.17	38.5 ± 2.17	**85.47 ± 0.12**	1
$\alpha = 10$	91.61 ± 0.19	91.25 ± 0.17	68.58 ± 0.13	**91.76 ± 0.13**	1
$\alpha = 1$	86.43 ± 0.38	86.13 ± 0.21	72.65 ± 0.37	**87.17 ± 0.24**	5
$\alpha = 10$	91.88 ± 0.23	91.66 ± 0.11	86.36 ± 0.15	**92.14 ± 0.22**	5

total 300 epochs and set the batch size to 128 on every worker. The learning rate is set to 5e-3 for single worker and warmed up in the first 5 epochs and decayed by a factor of 10 at 150 and 250-th epoch. All experiments were repeated three times with different seeds. For NEOLITHIC, we set $R = 2$. Following previous works [60], we use top-k compressor with different compression ratio to evaluate the performance of the aforementioned methods. As shown in Figure 1 (left), Figure 3, and Table 2, NEOLITHIC consistently outperforms other compression methods and reach the similar performance to PSGD. It is worth noting that EF21-SGD, while guaranteed to converge with milder assumptions than ours, does not provide competitive performance in deep learning tasks listed in Tables 2, 3, and 4. We find our reported result for EF21-SGD is consistent with Fig. 7 (the left plot) in [23].

Performance with heterogeneous data. We simulate data heterogeneity among nodes via a Dirichlet distribution-based partitioning with parameter α controlling the data heterogeneity. The training data for a particular class tends to concentrate in a single node as $\alpha \to 0$, i.e. becoming more heterogeneous, while the homogeneous data distribution is achieved as $\alpha \to \infty$. We test $\alpha = 1$ and 10 in Table 4 for all compared methods as corresponding to a setting with high/low heterogeneity.

Effects of accumulation rounds. We also empirically evaluate the performance of NEOLITHIC with different choice of parameter R in deep learning tasks. NEOLITHIC have slightly performance degradation in both compression scenarios as R scales up. We conjecture that the gradient accumulation step, which amounts to using large-batch samples in gradient evaluation, can help in the optimization and training stage as proved in this paper, but it may hurt the generalization performance. We recommend using NEOLITHIC in applications that are friendly to large-batch training.

Table 5: Effects of round numbers for CIFAR-10 dataset with ResNet-18

ROUNDS	2	3	4	5
NEOLITHIC (5%)	94.63 ± 0.09	93.32 ± 0.08	92.55 ± 0.12	91.48 ± 0.18
NEOLITHIC (1%)	94.155 ± 0.10	93.15 ± 0.11	92.27 ± 0.08	91.32 ± 0.12

D More Details of Table 1

There exist mismatches between some rates listed in Table 1 and those established in literature. The mismatches exist because

1. we have strengthened the vanilla rates by relaxing their restrictive assumptions (say, Double-Squeeze) or uncovering the hidden terms (say, CSER);

2. we have extended the vanilla rates to the same setting as NEOLITHIC (say, extend MEM-SGD to non-convex and smooth setting, or transform QSGD to the distributed setting).

With these modifications, these baseline algorithms can be compared with NEOLITHIC in a fair manner. Next we clarify each modification one by one.

D.1 Q-SGD

There is a slight inconsistency between the original rate in [30] and the one listed Table 1 due to the following reasons:

1. We noticed that the rate stated in [30, Corollary 3] is not optimal following [30, Theorem 2] since the authors set the learning rate as $\theta\sqrt{M/T}$ where θ is a universal constant. The rate in [30, Corollary 3] can be slightly improved in terms of σ^2, b^2, $\Delta = f(x^{(0)}) - \min_x f(x)$ by involving them into the learning rate. We manually optimize the learning rate by choosing $\Theta((L + (T(1+\omega)L\sigma^2/\Delta M)^{1/2} + (\omega LTb^2/n\Delta)^{1/2})^{-1})$.
2. M is the total mini-batch size on all workers in Q-SGD (see the paragraph of contribution, page 2, [30]). For a fair comparison with other methods in Table 1, We set $M = n$ to make the number of total gradient queries per iteration equivalent in all algorithms.

D.2 CSER

While [66, Corollary 1] does not have a $O(1/T)$ term in the convergence rate, it does impose another condition $T \gg n$ to the convergence statement. If $T \gg n$ holds, then $1/\sqrt{nT} \gg 1/T$ and hence the $1/T$ term is dominated by $1/\sqrt{nT}$ and thus hidden in the notation $O(\cdot)$. However, the condition does not appear in the convergence theorem of NEOLITHIC. To conduct a fair comparison, we have to remove condition $T \gg n$ from its convergence theorem, which thus incurs additional $O(1/T)$ term in the rate accordingly. The rate we provided in Table 1 is hence more precise than that in [66].

D.3 Double-Squeeze

The original rate of Double-Squeeze established in [60] is based on an unrealistic assumption that accumulated compression errors are bounded by unknown ϵ (see [60, Assumption 1.3]). This makes its rate incomparable with other methods. However, we can remove the unrealistic assumption and easily derive a comparable bound where ϵ can be explicitly replaced with $O(G/\delta^2)$. We plug $O(G/\delta^2)$ into [60, Corollary 2] to get the rate listed in our Table 1.

To this end, we follow the notations of [60] to derive explicit upper bounds for server/worker compression errors $\mathbb{E}[\|\boldsymbol{\delta}_t\|]$ and $\mathbb{E}[\|\boldsymbol{\delta}_t^{(i)}\|]$ which, combined with [60, Corollary 2], leads to the rate in our Table 1. We consider compressors Q (in server) and Q_i (in worker i) utilized are δ-contractive. We use $\boldsymbol{g}_t^{(i)}$ to indicate local (stochastic) gradient.

Bound of local compression error: $\mathbb{E}[\|\boldsymbol{\delta}_t^{(i)}\|] = O(G/\delta)$. By δ-contraction and Young's inequality, we have for any $\rho > 0$ that

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{\delta}_t^{(i)}\|^2] =& \mathbb{E}[\|\boldsymbol{v}_t^{(i)} - Q_i(\boldsymbol{v}_t^{(i)})\|^2] \\ \leq& (1-\delta)\mathbb{E}[\|\boldsymbol{v}_t^{(i)}\|^2] = (1-\delta)\mathbb{E}[\|\boldsymbol{g}_t^{(i)} + \boldsymbol{\delta}_{t-1}^{(i)}\|^2] \\ \leq& (1+\rho)(1-\delta)\mathbb{E}[\|\boldsymbol{g}_t^{(i)}\|^2] + (1+1/\rho)(1-\delta)\mathbb{E}[\|\boldsymbol{\delta}_{t-1}^{(i)}\|^2] \end{aligned} \tag{51}$$

Iterating (51) for $t, t-1, \ldots, 0$ and noting $\boldsymbol{\delta}_0^{(i)} = 0$, we reach

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{\delta}_t^{(i)}\|^2] \leq& (1+\rho)(1-\delta)\sum_{s=1}^{t}(1+1/\rho)^{t-s}(1-\delta)^{t-s}\mathbb{E}[\|\boldsymbol{g}_s^{(i)}\|^2] \\ \leq& \frac{(1+\rho)(1-\delta)G^2}{1-(1+1/\rho)(1-\delta)} \end{aligned} \tag{52}$$

where the last inequality holds because $\mathbb{E}[\|\boldsymbol{g}_s^{(i)}\|^2] \leq G^2$ for all $1 \leq s \leq t$. Here one must choose $\rho = \Omega(1/\delta)$ to avoid the explosion of the upper bound, which leads to $\mathbb{E}[\|\boldsymbol{\delta}_t^{(i)}\|^2] = O(G^2/\delta^2)$. Therefore, by Jessen's inequality, we have $\mathbb{E}[\|\boldsymbol{\delta}_t^{(i)}\|] \leq \sqrt{\mathbb{E}[\|\boldsymbol{\delta}_t^{(i)}\|^2]} = O(G/\delta)$.

Bound of global compression error: $\mathbb{E}[\|\boldsymbol{\delta}_t\|] = O(G/\delta^2)$**.** By δ-contraction and Young's inequality, we have for any $\rho > 0$ that

$$\begin{aligned}
\mathbb{E}[\|\boldsymbol{\delta}_t\|^2] =& \mathbb{E}[\|\boldsymbol{v}_t - Q(\boldsymbol{v}_t)\|^2] \\
\leq& (1-\delta)\mathbb{E}[\|\boldsymbol{v}_t\|^2] = (1-\delta)\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n Q_i(\boldsymbol{v}_t^{(i)}) + \boldsymbol{\delta}_{t-1}\right\|^2\right] \\
\leq& (1-\delta)(1+\rho)\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n Q_i(\boldsymbol{v}_t^{(i)})\right\|^2\right] + (1+1/\rho)(1-\delta)\mathbb{E}[\|\boldsymbol{\delta}_{t-1}\|^2]. \qquad (53)
\end{aligned}$$

Again by Young's inequality and δ-contraction, we have

$$\begin{aligned}
\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n Q_i(\boldsymbol{v}_t^{(i)})\right\|^2\right] \leq& 2\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n Q_i(\boldsymbol{v}_t^{(i)}) - \boldsymbol{v}_t^{(i)}\right\|^2\right] + 2\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{v}_t^{(i)}\right\|^2\right] \\
\leq& \frac{2-\delta}{n}\sum_{i=1}^n \mathbb{E}[\|\boldsymbol{v}_t^{(i)}\|^2] = O(G^2/\delta^2) \qquad (54)
\end{aligned}$$

where the last identity is because the upper bound of $\mathbb{E}[\|\boldsymbol{v}_t^{(i)}\|^2]$ can be obtained by following the derivation in (51) and (52). Taking $\rho = 2(1-\delta)/\delta = O(1/\delta)$ in (53) and using (54), we reach an inequality taking a form like

$$\mathbb{E}[\|\boldsymbol{\delta}_t\|^2] \leq (1-\delta/2)\mathbb{E}[\|\boldsymbol{\delta}_{t-1}\|^2] + O(G^2/\delta^3). \qquad (55)$$

Iterating (55) similarly, we easily reach $\mathbb{E}[\|\boldsymbol{\delta}_t\|^2] = O(G^2/\delta^4)$ and thus $\mathbb{E}[\|\boldsymbol{\delta}_t\|] = O(G/\delta^2)$.

D.4 MEM-SGD

We notice that only the rate for strongly convex problems is established in the original MEM-SGD paper [56]. To compare it fairly with NEOLITHIC, we derive its convergence rate in the non-convex setting by ourselves.

The main recursion of MEM-SGD is

$$\mathbf{p}_t^i = \eta\nabla f_i(\mathbf{x}_t, \xi_t^i) + \mathbf{e}_t^i, \quad \mathbf{x}_{t+1} = \mathbf{x}_t - \frac{1}{n}\sum_{i=1}^n Q_i(\mathbf{p}_t^i), \quad \mathbf{e}_{t+1}^i = \mathbf{p}_t^i - Q_i(\mathbf{p}_t^i).$$

The key steps of our derivation are listed as follows.

1. Following the similar argument to (51) and (52), we can bound the compression error as $\mathbb{E}[\|\mathbf{e}_t^i\|^2] = O(\eta^2 G^2/\delta^2)$. Note that here η^2 appears since compression is conducted after multiplying the learning rate η.
2. The recursion formula of MEM-SGD is $\mathbf{y}_{t+1} = \mathbf{y}_t - \frac{\eta}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_t, \xi_t^i)$ with $\mathbf{y}_t \triangleq \mathbf{x}_t - \frac{1}{n}\sum_{i=1}^n \mathbf{e}_t^i$. In fact, one can easily check that

$$\begin{aligned}
\mathbf{x}_{t+1} =& \mathbf{x}_t - \frac{1}{n}\sum_{i=1}^n Q_i(\mathbf{p}_t^i) \\
=& \mathbf{x}_t - \frac{1}{n}\sum_{i=1}^n (\mathbf{p}_t^i - \mathbf{e}_{t+1}^i) = \mathbf{x}_t - \frac{1}{n}\sum_{i=1}^n (\eta\nabla f_i(\mathbf{x}_t, \xi_t^i) + \mathbf{e}_t^i - \mathbf{e}_{t+1}^i) \\
=& \mathbf{y}_t - \frac{\eta}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_t, \xi_t^i) + \frac{1}{n}\sum_{i=1}^n \mathbf{e}_{t+1}^i.
\end{aligned}$$

Table 6: Comparison between between Q-SGD [30], VR-MARINA [61], and SASHA-MVR [25]. To explicitly clarify the influence of different compression strategies, we keep the stochastic gradient variance σ^2, data heterogeneity bound b^2 , mini-batch size B (used in [30, 25]), probability of conducting uncompressed communication p (used in [61]), but omit smoothness constant L, and initialization gap $f(x^{(0)}) - f^\star$ in the below results.

Algorithm	**#Communication**	**#Gradient Query per Worker**
Q-SGD [30]	$O\left(\frac{1}{nB\epsilon^4}\left((1+\omega)\sigma^2+\omega b^2\right)\right)$	$B\times$#Communication
VR-MARINA [25]	$O\left(\frac{1}{\epsilon^2}\left(1+\sqrt{\frac{1-p}{pn}\left(\omega+\frac{1+\omega}{\max\{1,\frac{\sigma^2}{n\epsilon^2}\}}\right)}\right)\right)$	$\max\{1,\frac{\sigma^2}{n\epsilon^2}\}\times$#Communication
DASHA-MVR [61]	$O\left(\frac{1}{\epsilon^2}\left(1+\frac{\sigma}{n\epsilon B^{3/2}}+\frac{\sigma^2}{\epsilon^2 B}\right)\right)$	$B\times$#Communication

3. Following the derivation of (34), one can obtain

$$\mathbb{E}[f(\mathbf{y}_{t+1})] - \mathbb{E}[f(\mathbf{y}_t)] \leq 2\eta L^2 \mathbb{E}[\|\mathbf{y}_t - \mathbf{x}_t\|^2] - \frac{\eta(1-\eta L)}{2}\mathbb{E}[\|\nabla f(\mathbf{x}_t)\|^2] + \frac{\eta^2 L\sigma^2}{2n}. \tag{56}$$

Setting $\eta \leq \frac{1}{2L}$ such that $\frac{\eta(1-\eta L)}{2} \geq \frac{\eta}{4}$ and rearranging (56), we have

$$\mathbb{E}[\|\nabla f(\mathbf{x}_t)\|^2] \leq \frac{4(\mathbb{E}[f(\mathbf{y}_t)] - \mathbb{E}[f(\mathbf{y}_{t+1})])}{\eta} + \frac{2\eta L\sigma^2}{n} + 8L^2\mathbb{E}[\|\mathbf{y}_t - \mathbf{x}_t\|^2]. \tag{57}$$

4. By the definition of $\mathbf{y}_t$ and step 1., we have

$$\mathbb{E}[\|\mathbf{y}_t - \mathbf{x}_t\|^2] \leq \frac{1}{n}\sum_{i=1}^n \mathbb{E}[\|\mathbf{e}_t^i\|^2] = O(\eta^2 G^2/\delta^2). \tag{58}$$

Averaging (57) with (58) plugged into, we reach

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(\mathbf{y}_t)\|^2] \leq O\left(\frac{\mathbb{E}[f(\mathbf{x}_0)] - f^\star}{\eta} + \frac{\eta L\sigma^2}{n} + \frac{\eta^2 L^2 G^2}{\delta^2}\right). \tag{59}$$

Setting the learning rate $\eta = (2L + (\frac{L\sigma^2}{n\Delta})^{1/2} + (\frac{L^2G^2}{\delta^2\Delta})^{1/3})^{-1}$ in (59) leads to the rate we listed in Table 1.

D.5 Comparison with More Algorithms

We supplement the comparison between Q-SGD [30], VR-MARINA [25], and DASHA-MVR [61] in Table 6. The results are compared in terms of the communication/query complexity to reach $\mathbb{E}[\|\nabla f(x)\|^2] \leq \epsilon$ for a sufficiently small ϵ.

Several additional comments are as follows:

1. All three algorithms utilize unidirectional, unbiased, and independent compressors.
2. All algorithms conduct an imbalanced number of compressed communications and of gradient queries. We therefore list the communication and gradient query complexity separately in Table 6. The result of Q-SGD is slightly tuned by us, see the argument in Appendix D.1.
3. MARINA (with $O(1/\epsilon^2)$) and SASHA (with $O(1/\epsilon^3)$) have better communication complexity than QSGD (with $O(1/\epsilon^4)$) when ϵ is sufficiently small.