CANITA: Faster Rates for Distributed Convex Optimization with Communication Compression

Anonymous Author(s)
Affiliation
Address
email

Abstract

Due to the high communication cost in distributed and federated learning, methods relying on compressed communication are becoming increasingly popular. Besides, the best theoretical and/or practical performing of gradient-type methods invariably rely on some form of acceleration/momentum to reduce the number of iterations, e.g., Nesterov's accelerated gradient descent [25] and Adam [11]. In order to enjoy the benefits of both communication compression and convergence acceleration, we propose a *compressed and accelerated* gradient method called CANITA for distributed optimization. In particular, CANITA achieves the *first accelerated rate* $O\Big(\sqrt{\Big(1+\sqrt{\frac{\omega^3}{n}}\Big)\frac{L}{\epsilon}}+\omega\big(\frac{1}{\epsilon}\big)^{\frac{1}{3}}\Big)$ which improves upon the state-of-the-art non-accelerated rate $O\Big((1+\frac{\omega}{n})\frac{L}{\epsilon}+\frac{\omega^2+n}{\omega+n}\frac{1}{\epsilon}\Big)$ of DIANA [6, 9] for distributed general convex problems, where ϵ is the target error $f(x^T)-f(x^*)\leq\epsilon$, L is the smooth parameter of f, n is the number of machines/devices, and ω is the compression parameter (larger ω means more compression can be applied, and no compression implies $\omega=0$). Our results show that as long as the number of devices n is large (often true in distributed/federated learning) or the compression ω is not very high, then CANITA achieves the faster convergence rate $O\big(\sqrt{\frac{L}{\epsilon}}\big)$, i.e., the number of communication rounds is $O\big(\sqrt{\frac{L}{\epsilon}}\big)$ (vs. $O\big(\frac{L}{\epsilon}\big)$ achieved by previous works). As a result, CANITA enjoys the advantages of both compression (compressed communication in each round) and acceleration (much fewer communication rounds).

1 Introduction

With the proliferation of edge devices such as mobile phones, wearables and smart home devices comes an increase in the amount of data rich in potential information which can be mined for the benefit of the users. One of the approaches of turning the raw data into information is via federated learning [12, 22], where typically a single global supervised model is trained in a massively distributed manner over a network of heterogeneous devices.

Training supervised distributed/federated learning models is typically performed by solving an optimization problem of the form

$$\min_{x\in\mathbb{R}^d}\Big\{f(x):=\frac{1}{n}\sum_{i=1}^{n}f_i(x)\Big\}, \tag{1}$$

where n denotes the number of devices/machines/workers and $f_i:\mathbb{R}^d\to\mathbb{R}$ is the smooth loss associated with data stored on device i.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

In distributed and federated learning problems (1), communication of messages across a network forms the bottleneck of the training system. It is thus very important to devise novel strategies for reducing the communication complexity. One of the most common strategies is communication compression [27, 1, 29, 5, 23, 6, 19, 18]. This strategy is trying to reduce the size of communicated messages, saving precious time spent in each communication round, and hoping that this will not increase the total number of communication rounds.

1.1 Methods with compressed communication

Indeed, several recent theoretical results suggest that by combining an appropriate (typically randomized) compression operator with a suitably designed gradient-type method, one can obtain improvement in the total communication complexity over comparable baselines not performing any compression. For instance, this is the case for distributed compressed gradient descent (CGD) [1, 10, 5, 18] and distributed CGD methods which employ variance reduction to tame the variance introduced by compression [4, 23, 6, 18, 3].

While in the case of CGD compression leads to a decrease in the size of communicated messages per communication round, it usually leads to an increase in the number of communication rounds. However, the number of communication rounds for certain methods [1, 23, 6, 19, 18] actually is unaffected up to a certain level of compression when the variance induced by compression (i.e,, ω in Definition 2) is smaller than a certain threshold. They also prove that this threshold can be very large (e.g., related to the number of machines/devices n), which means that massive reduction is often possible in the number of communicated bits without this having any adverse effect on the number of communication rounds.

1.2 Methods with acceleration/momentum

Encouraged by the theoretical success of CGD methods which can decrease the size of communicated messages, in this paper we further enhance CGD methods with acceleration/momentum which can provably lead to a decrease of the number of communication rounds. Our goal is to enjoy the advantages of both compression (compressed communication in each round) and acceleration (much fewer communication rounds).

The acceleration/momentum of gradient-type methods is widely-studied in standard optimization problems. See e.g., [24, 25, 14, 21, 2, 15, 13, 17, 16]. Especially, people usually use Adam [11] which also adopts momentum steps for training machine learning models in practice.

Thus a natural question is that "*whether the gradient-type methods with gradient compression can still theoretically enjoy the benefit of acceleration/momentum*"? However, there is very limited work to address this key question. The first successful combination of gradient compression and acceleration/momentum is recently proposed by Li et al. [19]. However, Li et al. [19] only provide the theoretical results for strongly convex problems. After the initial work [19], to the best of our knowledge, we only aware two other works [30, 26] for addressing this question. However, both of them [30, 26] still only focus on the strongly convex setting. Thus this line of research is still very restricted. For instance, the well-known logistic regression problem is convex but not strongly convex. Also the modulus of strong convexity usually is not known even if the problems are strongly convex.

1.3 Our contributions

In this paper, we push forward the benefits of compression and acceleration to the distributed general convex setting, i.e., the objective function is convex, while previous work [19] requires it to be strongly convex. Concretely, we propose a compressed accelerated gradient method called CANITA (described in Algorithm 1) for distributed convex optimization with communication compression. In particular, CANITA can be seen as a combination of accelerated ANITA method [16] and compressed DIANA method [23]. Now, we highlight the following results achieved by our CANITA:

- For general convex problems, CANITA obtains the *first accelerated rate* $O\Big(\sqrt{\Big(1+\sqrt{\frac{\omega^3}{n}}\Big)\frac{L}{\epsilon}} + \omega\big(\frac{1}{\epsilon}\big)^{\frac{1}{3}}\Big)$ which improves upon the state-of-the-art result $O\Big((1+\frac{\omega}{n})\frac{L}{\epsilon} + \frac{\omega^2+n}{\omega+n}\frac{1}{\epsilon}\Big)$ of DIANA [9]. See Table 1 for more comparisons.

Table 1: Convergence rates for finding an ϵ-solution $\mathbb{E}[f(x^T) - f(x^*)] \leq \epsilon$ of distributed problem (1)

Algorithms	Strongly convex [1]	General convex	Remark
QSGD [1]	—	$O\left(\frac{L}{\epsilon} + \frac{\omega G^2}{n}\frac{1}{\epsilon^2}\right)$ [2]	✓compression, × acceleration
DIANA [23]	$O\left(\left(\left(1+\frac{\omega}{n}\right)\kappa + \omega\right)\log\frac{1}{\epsilon}\right)$	—	✓compression, × acceleration
DIANA [6]	$O\left(\left(\left(1+\frac{\omega}{n}\right)\kappa + \omega\right)\log\frac{1}{\epsilon}\right)$	$O\left(\left(1+\frac{\omega}{n}\right)\frac{L}{\epsilon} + \frac{\omega}{\epsilon}\right)$	✓compression, × acceleration
DIANA [9]	—	$O\left(\left(1+\frac{\omega}{n}\right)\frac{L}{\epsilon} + \frac{\omega^2+n}{\omega+n}\frac{1}{\epsilon}\right)$	✓compression, × acceleration
ADIANA [19]	$O\left(\left(\sqrt{\kappa} + \sqrt{\left(\frac{\omega}{n}+\sqrt{\frac{\omega}{n}}\right)\omega\kappa} + \omega\right)\log\frac{1}{\epsilon}\right)$	—	✓compression, ✓acceleration
CANITA (this paper)	—	$O\left(\sqrt{\left(1+\sqrt{\frac{\omega^3}{n}}\right)\frac{L}{\epsilon}} + \omega\left(\frac{1}{\epsilon}\right)^{\frac{1}{3}}\right)$	✓compression, ✓acceleration

• For strongly convex problems, Li et al. [19] show that if the number of devices/machines n is large or the compression ω is not very high, i.e., $\omega \leq n^{1/3}$, then ADIANA enjoys the benefits of both compression and acceleration (i.e., $\sqrt{\kappa}\log\frac{1}{\epsilon}$ of ADIANA vs. $\kappa\log\frac{1}{\epsilon}$ of previous works). In this paper, we consider the general convex setting and show that the proposed CANITA also enjoys the benefits of both compression and acceleration. Similarly, if $\omega \leq n^{1/3}$ (i.e., lots of devices, or not high compression), then CANITA achieves the accelerated rate $\sqrt{\frac{L}{\epsilon}}$ vs. $\frac{L}{\epsilon}$ of previous works. Note that the second term $\left(\frac{1}{\epsilon}\right)^{\frac{1}{3}}$ is a lower order term compared with the first term $\sqrt{\frac{1}{\epsilon}}$ and it can also be removed if $\omega \leq \sqrt{L}\left(\frac{1}{\epsilon}\right)^{\frac{1}{6}}$ (i.e., small target error, or not high compression).

• The proof analysis of CANITA is substantially different than ADIANA [19] since there is no strong convexity for general convex problems. Moreover, the theoretical rate for strongly convex case is linear $O(\log\frac{1}{\epsilon})$ while it is sublinear $O(\frac{1}{\epsilon})$ or $O\left(\sqrt{\frac{1}{\epsilon}}\right)$ (accelerated) for general convex case. Thus we hope that our first analysis for accelerating general convex case can provide new insights and shed some light on future work in this line of research (compression + acceleration).

2 Preliminaries

Notation: Let $[n]$ denote the set $\{1, 2, \cdots, n\}$ and $\|\cdot\|$ denote the Euclidean norm for a vector and the spectral norm for a matrix. Let $\langle u, v\rangle$ denote the inner product of two vectors u and v. We use $O(\cdot)$ and $\Omega(\cdot)$ to hide the absolute constant. We will write $x^* := \arg\min_{x\in\mathbb{R}^d} f(x)$.

Definition 1 *A point $\widehat{x}$ is called an ϵ-solution for distributed problem* (1) *if* $\mathbb{E}[f(\widehat{x}) - f(x^*)] \leq \epsilon$.

2.1 Compression operators

We now introduce the notion of a randomized compression operator which is used to compress the gradients. Also see e.g., [1, 10, 6, 18, 19].

Definition 2 (Compression operator) *A randomized map $\mathcal{C} : \mathbb{R}^d \mapsto \mathbb{R}^d$ is an ω-compression operator if* $\forall x \in \mathbb{R}^d$,

$$\mathbb{E}[\mathcal{C}(x)] = x, \quad \mathbb{E}[\|\mathcal{C}(x) - x\|^2] \leq \omega\|x\|^2. \tag{2}$$

In particular, no compression ($\mathcal{C}(x) \equiv x$) implies $\omega = 0$.

[1] In this strongly convex column, $\kappa := \frac{L}{\mu}$ denotes the condition number, where L is the smooth parameter and μ is the strongly convex parameter.

[2] Here QSGD [1] needs an additional bounded gradient assumption, i.e., $\|\nabla f_i(x)\|^2 \leq G^2, \forall i \in [n], x \in \mathbb{R}^d$.

Note that the conditions (2) is satisfied by many practical compressions. We give a few examples as follows.

Example 1 (Random sparsification): Given $x \in \mathbb{R}^d$, the random-k sparsification operator is defined by

$$\mathcal{C}(x) := \frac{d}{k}(\xi_k \odot x),$$

where $\odot$ denotes the Hadamard (element-wise) product and $\xi_k \in \{0,1\}^d$ is a uniformly random binary vector with k nonzero entries ($\|\xi_k\|_0 = k$). This random-k sparsification operator $\mathcal{C}$ satisfies (2) with $\omega = \frac{d}{k} - 1$. Indeed, no compression $k = d$ implies $\omega = 0$.

Example 2 (Quantization): Given $x \in \mathbb{R}^d$, the (p, s)-quantization operator is defined by

$$\mathcal{C}(x) := \text{sign}(x) \cdot \|x\|_p \cdot \frac{1}{s} \cdot \xi_s,$$

where $\xi_s \in \mathbb{R}^d$ is a random vector with i-th element

$$\xi_s(i) := \begin{cases} l+1, & \text{with probability } \frac{|x_i|}{\|x\|_p} s - l \\ l, & \text{otherwise} \end{cases},$$

where the level l satisfies $\frac{|x_i|}{\|x\|_p} \in [\frac{l}{s}, \frac{l+1}{s}]$. The probability is chosen so that $\mathbb{E}[\xi_s(i)] = \frac{|x_i|}{\|x\|_p} s$. This (p, s)-quantization operator $\mathcal{C}$ satisfies (2) with $\omega = 2 + \frac{d^{1/p} + d^{1/2}}{s}$. In particular, QSGD [1] used $p = 2$ (i.e., $(2, s)$-quantization) and proved that the expected sparsity of $\mathcal{C}(x)$ is $\mathbb{E}[\|\mathcal{C}(x)\|_0] = O\big(s(s + \sqrt{d})\big)$.

2.2 Assumption

To show the convergence results, we assume the following standard smoothness assumption for functions f_is in (1).

Assumption 1 *Functions $f_i : \mathbb{R}^d \to \mathbb{R}$ are convex and L-smooth such that*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\| \tag{3}$$

for some $L \geq 0$ and all $i \in [n]$.

It is easy to see that $f(x) = \frac{1}{n}\sum_{i=1}^n f_i(x)$ is also L-smooth under Assumption 1.

3 The CANITA Algorithm

In this section, we describe the compressed and accelerated gradient method (called CANITA) in Algorithm 1 for solving problem (1) in distributed and federated learning. In particular, CANITA can be seen as a combination of accelerated ANITA method [16] and compressed DIANA method [23]. Concretely, CANITA adopts the momentum/acceleration steps (see Line 3 and 10 of Algorithm 1) similar to ANITA [16], and adopts the shifted compression steps for each machine/device (see Line 5 and 8 of Algorithm 1) similar to DIANA [23]. As a result, we prove that CANITA can enjoy the benefits of both methods *simultaneously*, i.e.,, *acceleration* from ANITA and *compression* from DIANA.

Although CANITA is a kind of combination of ANITA [16] and DIANA [23, 6, 9] from the algorithmic perspective, the proof analysis of CANITA is totally different. Compared with ANITA [16], CANITA needs to deal with the extra compression of shifted local gradients among the distributed network. Thus the obtained gradient estimator g^k in Line 8 of Algorithm 1 is totally different and more complicated than the one in ANITA. Compared with DIANA [23, 6, 9], the extra momentum steps in Line 3 and 10 of Algorithm 1 make the analysis of CANITA more complicated than DIANA and also lead to an accelerated rate $O(\sqrt{\frac{L}{\epsilon}})$ than the non-accelerated rate $O(\frac{L}{\epsilon})$ in DIANA. Finally, compared with ADIANA [19], the analysis of CANITA is also substantially different since CANITA cannot exploit the strong convexity rather than ADIANA. Also see the last bullet point in Section 1.3 our contributions for more discussions.

Algorithm 1 Distributed compressed accelerated ANITA method (CANITA)

Input: initial point x^0, $\{h_i^0\}_{i=1}^n$, $h^0 = \frac{1}{n}\sum_{i=1}^n h_i^0$, parameters $\{p_t\}, \{\theta_t\}, \{\alpha_t\}, \{\eta_t\}$

$w^0 = \bar{x}^0 = x^0$

for $t = 0, 1, 2, \ldots$ **do**

  $\underline{x}^t = \theta_t x^t + (1-\theta_t) w^t$

  for all machines $i = 1, 2, \ldots, n$ **do in parallel**

    Compress shifted local gradient $\mathcal{C}_i^t(\nabla f_i(\underline{x}^t) - h_i^t)$ and send to the server

    Update local shift $h_i^{t+1} = h_i^t + \alpha_t \mathcal{C}_i^t(\nabla f_i(w^t) - h_i^t)$

  end for

  Aggregate received compressed local gradient information:

$$g^t = \frac{1}{n}\sum_{i=1}^n \mathcal{C}_i^t(\nabla f_i(\underline{x}^t) - h_i^t) + h^t$$

$$h^{t+1} = h^t + \alpha_t \frac{1}{n}\sum_{i=1}^n \mathcal{C}_i^t(\nabla f_i(w^t) - h_i^t)$$

  Perform update step:

$$x^{t+1} = x^t - \frac{\eta_t}{\theta_t} g^t$$

  $\bar{x}^{t+1} = \theta_t x^{t+1} + (1-\theta_t) w^t$

  $w^{t+1} = \begin{cases} \bar{x}^{t+1}, & \text{with probability } p_t \\ w^t, & \text{with probability } 1 - p_t \end{cases}$

end for

4 Convergence Results for CANITA Algorithm

In this section, we provide the convergence results for CANITA (Algorithm 1). We first present the main convergence theorem of CANITA for distributed general convex problems. Then we provide a corollary for obtaining a detailed convergence rate. We will provide a proof sketch of the main Theorem 1 in the next Section 5. The full detailed proofs are deferred to the appendix.

Theorem 1 *Suppose that Assumption 1 holds. For any $\beta_t, \gamma_t > 0$, let $\eta_t \leq \frac{1}{L(1+\beta_t+4p_t\gamma_t(1+2p_t/\alpha_t))}$ and $\alpha_t \leq \frac{1}{1+\omega}$ for any $t \geq 0$, and let $\frac{2\omega}{\beta_t n} + 4p_t\gamma_t(1+\frac{2p_t}{\alpha_t}) \leq 1-\theta_t$, $\frac{(1-p_t\theta_t)\eta_t}{p_t\theta_t^2} \leq \frac{\eta_{t-1}}{p_{t-1}\theta_{t-1}^2}$, and $\left(\frac{\omega}{\beta_t n} + (1-\frac{\alpha_t}{2})\gamma_t\right)\frac{\eta_t}{\theta_t^2} \leq \frac{\gamma_{t-1}\eta_{t-1}}{\theta_{t-1}^2}$ for any $t \geq 1$, then the following equation holds for* CANITA *(Algorithm 1) for any iteration $t \geq 0$:*

$$\begin{aligned}
&\mathbb{E}\left[\Big(f(w^{t+1}) - f(x^*)\Big) + \frac{\gamma_t p_t}{Ln}\sum_{i=1}^n \|\nabla f_i(w^{t+1}) - h_i^{t+1}\|^2\right] \\
&\leq \frac{\theta_t^2 p_t}{\eta_t}\Bigg(\frac{(1-\theta_0 p_0)\eta_0}{\theta_0^2 p_0}\big(f(w^0) - f(x^*)\big) + \big(\frac{\omega}{\beta_0 n} + (1-\frac{\alpha_0}{2})\gamma_0\big)\frac{\eta_0}{\theta_0^2 Ln}\sum_{i=1}^n \|\nabla f_i(w^0) - h_i^0\|^2 \\
&\qquad + \frac{1}{2}\|x^0 - x^*\|^2\Bigg).
\end{aligned} \tag{4}$$

Remark: The probability p_t and stepsize η_t usually are fixed to some constants while the parameter θ_t is decreasing, e.g., $p_t \equiv \frac{1}{a_1}$, $\eta_t \equiv \frac{1}{a_2 L}$, $\eta_t = \frac{a_3}{t+a_4}$ for some factors a_1, a_2, a_3 and a_4. Note that the factors a_1, a_2, a_3 and a_4 may depend on compression parameter ω and the number of devices/machines n. Thus the right-hand-side of (4) will be the order of $O(\frac{L}{t^2})$. Then for finding an ϵ-solution such that $\mathbb{E}[f(w^{T+1}) - f(x^*)] \leq \epsilon$, the number of iterations/communication rounds of CANITA (Algorithm 1) is at most $T = O\left(\sqrt{\frac{L}{\epsilon}}\right)$.

Now, we formulate a corollary of Theorem 1 which lead to the detailed convergence results of CANITA by specifying the choice of parameters.

Corollary 1 *Suppose that Assumption 1 holds. Choosing the parameters of Algorithm 1 as follows:*

$$p_t \equiv \frac{1}{1+b}, \quad \theta_t = \frac{3(1+b)}{t+9(1+b+\omega)}, \quad \alpha_t \equiv \frac{1}{1+\omega}, \quad \eta_t \leq \frac{1}{L(\beta_t+3/2)}, \tag{5}$$

where b is chosen as $b = \min\left\{\omega, \sqrt{\frac{\omega(1+\omega)^2}{n}}\right\}$, and β_t is chosen as

$$\beta_t \equiv \beta = \frac{48\omega(1+\omega)(1+b+2(1+\omega))}{n(1+b)^2}, \ \ \forall t \geq 1, \tag{6}$$

and let $\beta_0 = \frac{9(1+b+\omega)^2}{(1+b)L}$ for $t = 0$. Then the following equation holds for CANITA *(Algorithm 1) for any iteration $T \geq 0$:*

$$\mathbb{E}\left[f(w^{T+1}) - f(x^*)\right] \leq O\left(\frac{(1+\sqrt{\omega^3/n})L}{T^2} + \frac{\omega^3}{T^3}\right). \tag{7}$$

According to (7)*, the number of iterations/communication rounds for* CANITA *(Algorithm 1) to find an ϵ-solution such that*

$$\mathbb{E}[f(w^{T+1}) - f(x^*)] \leq \epsilon$$

is at most

$$T = O\left(\sqrt{\left(1+\sqrt{\frac{\omega^3}{n}}\right)\frac{L}{\epsilon}} + \omega\left(\frac{1}{\epsilon}\right)^{\frac{1}{3}}\right).$$

5 Proof Sketch of Main Theorem 1

Now, we provide the proof sketch of Theorem 1. First, we need a useful Lemma 1 which connects the change of function value after a gradient update step.

Lemma 1 *Suppose that Assumption 1 holds. For any $\beta_t > 0$, the following equation holds for* CANITA *(Algorithm 1) for any iteration $t \geq 0$:*

$$\begin{aligned}\mathbb{E}[f(\bar{x}^{t+1})] \leq \mathbb{E}\Big[&f(\underline{x}^t) + \langle \nabla f(\underline{x}^t), \theta_t(x^* - x^t)\rangle + \frac{\theta_t^2}{2\eta_t}\big(\|x^t - x^*\|^2 - \|x^{t+1} - x^*\|^2\big) \\ &- \big(\frac{\theta_t^2}{2\eta_t} - \frac{L(1+\beta_t)\theta_t^2}{2}\big)\|x^{t+1} - x^t\|^2 + \frac{1}{2L\beta_t}\|\nabla f(\underline{x}^t) - g^t\|^2\Big]. \end{aligned} \tag{8}$$

Note that $\bar{x}^{t+1} - \underline{x}^t = \theta_t(x^{t+1} - x^t) = -\eta_t g^t$ according to the two momentum/interpolation steps of CANITA (see Line 3 and Line 10 of Algorithm 1) and the gradient update step (see Line 9 of Algorithm 1). The proof of Lemma 1 uses these relations and L-smoothness Assumption 1.

Next, we provide the following Lemma 2 to bound the last variance term $\mathbb{E}\|\nabla f(\underline{x}^t) - g^t\|^2$ in (8) of Lemma 1.

Lemma 2 *If g^k is defined in Line 8 of Algorithm 1 and the compression operator $\mathcal{C}(\cdot)$ satisfies* (2)*, we have*

$$\mathbb{E}[\|\nabla f(\underline{x}^t) - g^t\|^2] \leq \frac{2\omega}{n^2}\sum_{i=1}^n \|\nabla f_i(\underline{x}^t) - \nabla f_i(w^t)\|^2 + \frac{2\omega}{n^2}\sum_{i=1}^n \|\nabla f_i(w^t) - h_i^t\|^2. \tag{9}$$

This lemma is proved by using the definition of ω-compression operator (i.e., (2)). Then we also need a lemma to handle the last term in (9) of Lemma 2.

Lemma 3 *Suppose that Assumption 1 holds. According to the probabilistic update of w^{t+1} in Line 11 of Algorithm 1 and let $\alpha_t \leq \frac{1}{1+\omega}$, we have*

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(w^{t+1}) - h_i^{t+1}\|^2\right] \leq \left(1-\frac{\alpha_t}{2}\right)\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(w^t) - h_i^t\|^2 + \left(1+\frac{2p_t}{\alpha_t}\right)\frac{2p_t}{n}\sum_{i=1}^{n}\left\|\nabla f_i(\underline{x}^t) - \nabla f_i(w^t)\right\|^2 + 2p_t L^2\theta_t^2\big(1+\frac{2p_t}{\alpha_t}\big)\mathbb{E}[\|x^{t+1}-x^t\|^2]. \tag{10}$$

This lemma is proved by using the update of w^{t+1} (Line 11 of Algorithm 1) and h_i^{t+1} (Line 6 of Algorithm 1), the property of ω-compression operator (i.e., (2)), and L-smoothness Assumption 1.

To deal with the term $\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(\underline{x}^t) - \nabla f_i(w^t)\|^2$ in Lemmas 2 and 3, we need the following Lemma 4.

Lemma 4 *Suppose that Assumption 1 holds. For any $\underline{x}^t, w^t \in \mathbb{R}^d$, the following inequality holds:*

$$\frac{1}{n}\sum_{i=1}^{n}\left\|\nabla f_i(\underline{x}^t) - \nabla f_i(w^t)\right\|^2 \leq 2L(f(w^t) - f(\underline{x}^t) - \langle \nabla f(\underline{x}^t), w^t - \underline{x}^t\rangle). \tag{11}$$

This lemma directly follows from a standard result under L-smoothness Assumption 1 (see e.g., Lemma 1 of [15, 16]).

Finally, we also need the following Lemma 5 to connect the relation of $f(w^{t+1})$ and $f(\underline{x}^{t+1})$.

Lemma 5 *According to the probabilistic update of w^{t+1} in Line 11 of Algorithm 1, we have*

$$\mathbb{E}[f(w^{t+1})] = p_t\mathbb{E}[f(\bar{x}^{t+1})] + (1-p_t)\mathbb{E}[f(w^t)]. \tag{12}$$

The proof of Lemma 5 simply follows from the probabilistic update of w^{t+1} (see Line 11 of Algorithm 1).

Now, we combine Lemmas 1–5 to obtain the following key lemma, which describes the recursive form of the function value after an iteration.

Lemma 6 *Suppose that Assumption 1 holds. For any $\beta_t, \gamma_t > 0$, let $\eta_t \leq \frac{1}{L(1+\beta_t+4p_t\gamma_t(1+2p_t/\alpha_t))}$ and $\alpha_t \leq \frac{1}{1+\omega}$ for any $t \geq 0$, and let $\frac{2\omega}{\beta_t n} + 4p_t\gamma_t(1+\frac{2p_t}{\alpha_t}) \leq 1-\theta_t$ for any $t \geq 1$, then the following equation holds for* CANITA *(Algorithm 1) for any iteration $t \geq 0$:*

$$\begin{aligned}
&\mathbb{E}\left[f(w^{t+1}) - f(x^*) + \frac{\gamma_t p_t}{Ln}\sum_{i=1}^{n}\|\nabla f_i(w^{t+1}) - h_i^{t+1}\|^2\right] \\
&\leq \mathbb{E}\Bigg[(1-\theta_t p_t)\Big(f(w^t) - f(x^*)\Big) + \Big(\frac{\omega}{\beta_t n} + (1-\frac{\alpha_t}{2})\gamma_t\Big)\frac{p_t}{Ln}\sum_{i=1}^{n}\|\nabla f_i(w^t) - h_i^t\|^2 \\
&\qquad + \frac{\theta_t^2 p_t}{2\eta_t}\Big(\|x^t - x^*\|^2 - \|x^{t+1} - x^*\|^2\Big)\Bigg].
\end{aligned} \tag{13}$$

Proof of Lemma 6. First, we plug (9) of Lemma 2 into (8) of Lemma 1 to obtain

$$\begin{aligned}
\mathbb{E}[f(\bar{x}^{t+1})] \leq \mathbb{E}\Bigg[&f(\underline{x}^t) + \langle \nabla f(\underline{x}^t), \theta_t(x^* - x^t)\rangle + \frac{\theta_t^2}{2\eta_t}\big(\|x^t - x^*\|^2 - \|x^{t+1} - x^*\|^2\big) \\
&- \big(\frac{\theta_t^2}{2\eta_t} - \frac{L(1+\beta_t)\theta_t^2}{2}\big)\|x^{t+1} - x^t\|^2 \\
&+ \frac{\omega}{L\beta_t n^2}\sum_{i=1}^{n}\|\nabla f_i(\underline{x}^t) - \nabla f_i(w^t)\|^2 + \frac{\omega}{L\beta_t n^2}\sum_{i=1}^{n}\|\nabla f_i(w^t) - h_i^t\|^2\Bigg].
\end{aligned} \tag{14}$$

Then, we add (14) and (10) $\times \frac{\gamma_t}{L}$ to get

$$
\begin{aligned}
&\mathbb{E}\left[f(\bar{x}^{t+1}) + \frac{\gamma_t}{Ln}\sum_{i=1}^{n}\|\nabla f_i(w^{t+1}) - h_i^{t+1}\|^2\right] \\
&\leq \mathbb{E}\Bigg[f(\underline{x}^t) + \langle \nabla f(\underline{x}^t), \theta_t(x^* - x^t)\rangle + \frac{\theta_t^2}{2\eta_t}\big(\|x^t - x^*\|^2 - \|x^{t+1} - x^*\|^2\big) \\
&\qquad - \big(\frac{\theta_t^2}{2\eta_t} - \frac{L(1+\beta_t)\theta_t^2}{2}\big)\|x^{t+1} - x^t\|^2 + \frac{\omega}{L\beta_t n^2}\sum_{i=1}^{n}\|\nabla f_i(\underline{x}^t) - \nabla f_i(w^t)\|^2 \\
&\qquad + \Big(\frac{\omega}{\beta_t n} + (1 - \frac{\alpha_t}{2})\gamma_t\Big)\frac{1}{Ln}\sum_{i=1}^{n}\|\nabla f_i(w^t) - h_i^t\|^2 \\
&\qquad + \left(1 + \frac{2p_t}{\alpha_t}\right)\frac{2p_t\gamma_t}{Ln}\sum_{i=1}^{n}\big\|\nabla f_i(\underline{x}^t) - \nabla f_i(w^t)\big\|^2 + 2p_t\gamma_t L\theta_t^2\big(1 + \frac{2p_t}{\alpha_t}\big)\|x^{t+1} - x^t\|^2\Bigg] \\
&= \mathbb{E}\Bigg[f(\underline{x}^t) + \langle \nabla f(\underline{x}^t), \theta_t(x^* - x^t)\rangle + \frac{\theta_t^2}{2\eta_t}\big(\|x^t - x^*\|^2 - \|x^{t+1} - x^*\|^2\big) \\
&\qquad - \Big(\frac{\theta_t^2}{2\eta_t} - \frac{L(1+\beta_t)\theta_t^2}{2} - 2p_t\gamma_t L\theta_t^2\big(1 + \frac{2p_t}{\alpha_t}\big)\Big)\|x^{t+1} - x^t\|^2 \\
&\qquad + \Big(\frac{\omega}{\beta_t n} + (1 - \frac{\alpha_t}{2})\gamma_t\Big)\frac{1}{Ln}\sum_{i=1}^{n}\|\nabla f_i(w^t) - h_i^t\|^2 \\
&\qquad + \left(\frac{2\omega}{\beta_t n} + 4p_t\gamma_t\big(1 + \frac{2p_t}{\alpha_t}\big)\right)\frac{1}{2Ln}\sum_{i=1}^{n}\big\|\nabla f_i(\underline{x}^t) - \nabla f_i(w^t)\big\|^2\Bigg] \\
&\leq \mathbb{E}\Bigg[f(\underline{x}^t) + \langle \nabla f(\underline{x}^t), \theta_t(x^* - x^t)\rangle + \frac{\theta_t^2}{2\eta_t}\big(\|x^t - x^*\|^2 - \|x^{t+1} - x^*\|^2\big) \\
&\qquad + \Big(\frac{\omega}{\beta_t n} + (1 - \frac{\alpha_t}{2})\gamma_t\Big)\frac{1}{Ln}\sum_{i=1}^{n}\|\nabla f_i(w^t) - h_i^t\|^2 \\
&\qquad + \left(\frac{2\omega}{\beta_t n} + 4p_t\gamma_t\big(1 + \frac{2p_t}{\alpha_t}\big)\right)\frac{1}{2Ln}\sum_{i=1}^{n}\big\|\nabla f_i(\underline{x}^t) - \nabla f_i(w^t)\big\|^2\Bigg] \qquad (15) \\
&\leq \mathbb{E}\Bigg[f(\underline{x}^t) + \langle \nabla f(\underline{x}^t), \theta_t(x^* - x^t)\rangle + \frac{\theta_t^2}{2\eta_t}\big(\|x^t - x^*\|^2 - \|x^{t+1} - x^*\|^2\big) \\
&\qquad + \Big(\frac{\omega}{\beta_t n} + (1 - \frac{\alpha_t}{2})\gamma_t\Big)\frac{1}{Ln}\sum_{i=1}^{n}\|\nabla f_i(w^t) - h_i^t\|^2 \\
&\qquad + \frac{1-\theta_t}{2Ln}\sum_{i=1}^{n}\big\|\nabla f_i(\underline{x}^t) - \nabla f_i(w^t)\big\|^2\Bigg] \qquad (16) \\
&\leq \mathbb{E}\Bigg[f(\underline{x}^t) + \langle \nabla f(\underline{x}^t), \theta_t(x^* - x^t)\rangle + \frac{\theta_t^2}{2\eta_t}\big(\|x^t - x^*\|^2 - \|x^{t+1} - x^*\|^2\big) \\
&\qquad + \Big(\frac{\omega}{\beta_t n} + (1 - \frac{\alpha_t}{2})\gamma_t\Big)\frac{1}{Ln}\sum_{i=1}^{n}\|\nabla f_i(w^t) - h_i^t\|^2 \\
&\qquad + (1-\theta_t)\Big(f(w^t) - f(\underline{x}^t) - \langle \nabla f(\underline{x}^t), w^t - \underline{x}^t\rangle\Big)\Bigg] \qquad (17) \\
&= \mathbb{E}\Bigg[f(\underline{x}^t) + \langle \nabla f(\underline{x}^t), \theta_t(x^* - x^t)\rangle + \frac{\theta_t^2}{2\eta_t}\big(\|x^t - x^*\|^2 - \|x^{t+1} - x^*\|^2\big) \\
&\qquad + \Big(\frac{\omega}{\beta_t n} + (1 - \frac{\alpha_t}{2})\gamma_t\Big)\frac{1}{Ln}\sum_{i=1}^{n}\|\nabla f_i(w^t) - h_i^t\|^2 \\
&\qquad + (1-\theta_t)\big(f(w^t) - f(\underline{x}^t)\big) - \theta_t\langle \nabla f(\underline{x}^t), \underline{x}^t - x^t\rangle\big)\Bigg] \qquad (18)
\end{aligned}
$$

$$\leq \mathbb{E}\Big[(1-\theta_t)f(w^t) + \theta_t f(x^*) + \frac{\theta_t^2}{2\eta_t}\big(\|x^t - x^*\|^2 - \|x^{t+1} - x^*\|^2\big) + \Big(\frac{\omega}{\beta_t n} + (1 - \frac{\alpha_t}{2})\gamma_t\Big)\frac{1}{Ln}\sum_{i=1}^n \|\nabla f_i(w^t) - h_i^t\|^2\Big], \tag{19}$$

where (15) holds by letting $\eta_t \leq \frac{1}{L(1+\beta_t+4p_t\gamma_t(1+2p_t/\alpha_t))}$, (16) holds by letting $\frac{2\omega}{\beta_t n} + 4p_t\gamma_t(1 + \frac{2p_t}{\alpha_t}) \leq 1 - \theta_t$, (17) follows from (11), (18) holds since $\underline{x}^t = \theta_t x^t + (1-\theta_t)w^t$ (see Line 3 of Algorithm 1), and the last inequality (19) uses the convexity of f. Also note that (16) from (15) uses $\frac{2\omega}{\beta_t n} + 4p_t\gamma_t(1 + \frac{2p_t}{\alpha_t}) \leq 1 - \theta_t$, however this condition is only needed for $t \geq 1$, i.e., it is not needed for the case $t = 0$ since $\underline{x}^0 = w^0 = x^0$. The function and inner product terms will also perform the same result in the final (19) since $\underline{x}^0 = w^0 = x^0$. The proof of Lemma 6 is finished by adding (19) $\times$ p_t and (12) to obtain (13). □

Now, we are ready to prove the main convergence Theorem 1. According to Lemma 6, we know the change of function value after an iteration. By dividing (13) with $\frac{\theta_t^2 p_t}{\eta_t}$ for both sides, we obtain

$$\mathbb{E}\left[\frac{\eta_t}{\theta_t^2 p_t}\Big(f(w^{t+1}) - f(x^*)\Big) + \frac{\gamma_t\eta_t}{\theta_t^2 Ln}\sum_{i=1}^n \|\nabla f_i(w^{t+1}) - h_i^{t+1}\|^2\right]$$
$$\leq \mathbb{E}\Big[\frac{(1-\theta_t p_t)\eta_t}{\theta_t^2 p_t}\Big(f(w^t) - f(x^*)\Big) + \Big(\frac{\omega}{\beta_t n} + (1-\frac{\alpha_t}{2})\gamma_t\Big)\frac{\eta_t}{\theta_t^2 Ln}\sum_{i=1}^n \|\nabla f_i(w^t) - h_i^t\|^2 + \frac{1}{2}\Big(\|x^t - x^*\|^2 - \|x^{t+1} - x^*\|^2\Big)\Big]. \tag{20}$$

Then according to the following conditions on the parameters in Theorem 1:

$$\frac{(1-p_t\theta_t)\eta_t}{p_t\theta_t^2} \leq \frac{\eta_{t-1}}{p_{t-1}\theta_{t-1}^2} \text{ and } \Big(\frac{\omega}{\beta_t n} + (1-\frac{\alpha_t}{2})\gamma_t\Big)\frac{\eta_t}{\theta_t^2} \leq \frac{\gamma_{t-1}\eta_{t-1}}{\theta_{t-1}^2}, \ \forall t \geq 1. \tag{21}$$

This proof of main Theorem 1 is finished by telescoping (20) from $t = 1$ to T via (21) and maintaining the same inequality (20) for $t = 0$:

$$\mathbb{E}\left[\Big(f(w^{T+1}) - f(x^*)\Big) + \frac{\gamma_T p_T}{Ln}\sum_{i=1}^n \|\nabla f_i(w^{T+1}) - h_i^{T+1}\|^2\right]$$
$$\leq \frac{\theta_T^2 p_T}{\eta_T}\Big(\frac{(1-\theta_0 p_0)\eta_0}{\theta_0^2 p_0}\big(f(w^0) - f(x^*)\big) + \big(\frac{\omega}{\beta_0 n} + (1-\frac{\alpha_0}{2})\gamma_0\big)\frac{\eta_0}{\theta_0^2 Ln}\sum_{i=1}^n \|\nabla f_i(w^0) - h_i^0\|^2 + \frac{1}{2}\|x^0 - x^*\|^2\Big). \tag{22}$$

6 Conclusion

In this paper, we propose a compressed and accelerated gradient method CANITA for distributed convex optimization with communication compression. In particular, we show that CANITA enjoys the advantages of both compression (compressed communication in each round) and acceleration (much fewer communication rounds). There is very limited work on this line of research (compression + acceleration). Previous related works currently only focus on the strongly convex setting. Here we push forward the research line to the general convex setting, i.e., CANITA achieves the first accelerated rate (i.e., uses much fewer communication rounds) for general convex problems while maintaining the compressed communication among the distributed network. We hope that our first analysis for accelerating the general convex setting can provide new insights and shed some light on future work in this line of research. We leave some interesting future work for further improvements, e.g., benefits from multiple local update [22, 28, 8, 7], benefits from variance reduction [6, 18], and more general nonconvex setting [20, 3].

References

[1] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pages 1709–1720, 2017.

[2] Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1200–1205. ACM, 2017.

[3] Eduard Gorbunov, Konstantin Burlachenko, Zhize Li, and Peter Richtárik. MARINA: Faster non-convex distributed learning with compression. In *International Conference on Machine Learning*, 2021.

[4] Filip Hanzely, Konstantin Mishchenko, and Peter Richtárik. SEGA: variance reduction via gradient sketching. In *Advances in Neural Information Processing Systems 31*, pages 2082–2093, 2018.

[5] Samuel Horváth, Chen-Yu Ho, Ľudovít Horváth, Atal Narayan Sahu, Marco Canini, and Peter Richtárik. Natural compression for distributed deep learning. *arXiv preprint arXiv:1905.10988*, 2019.

[6] Samuel Horváth, Dmitry Kovalev, Konstantin Mishchenko, Sebastian Stich, and Peter Richtárik. Stochastic distributed learning with gradient quantization and variance reduction. *arXiv preprint arXiv:1904.05115*, 2019.

[7] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

[8] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local SGD on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 4519–4529. PMLR, 2020.

[9] Ahmed Khaled, Othmane Sebbouh, Nicolas Loizou, Robert M Gower, and Peter Richtárik. Unified analysis of stochastic gradient methods for composite convex and smooth optimization. *arXiv preprint arXiv:2006.11573*, 2020.

[10] Sarit Khirirat, Hamid Reza Feyzmahdavian, and Mikael Johansson. Distributed learning with compressed gradients. *arXiv preprint arXiv:1806.06573*, 2018.

[11] Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *The 3rd International Conference on Learning Representations*, 2014.

[12] Jakub Konečný, H. Brendan McMahan, Felix Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: strategies for improving communication efficiency. In *NIPS Private Multi-Party Machine Learning Workshop*, 2016.

[13] Dmitry Kovalev, Samuel Horváth, and Peter Richtárik. Don't jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop. In *Proceedings of the 31st International Conference on Algorithmic Learning Theory*, 2020.

[14] Guanghui Lan and Yi Zhou. An optimal randomized incremental gradient method. *arXiv preprint arXiv:1507.02000*, 2015.

[15] Guanghui Lan, Zhize Li, and Yi Zhou. A unified variance-reduced accelerated gradient method for convex optimization. In *Advances in Neural Information Processing Systems*, pages 10462–10472, 2019.

[16] Zhize Li. ANITA: An optimal loopless accelerated variance-reduced gradient method. *arXiv preprint arXiv:2103.11333*, 2021.

[17] Zhize Li and Jian Li. A fast Anderson-Chebyshev acceleration for nonlinear optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 1047–1057. PMLR, 2020.

[18] Zhize Li and Peter Richtárik. A unified analysis of stochastic gradient methods for nonconvex federated optimization. *arXiv preprint arXiv:2006.07013*, 2020.

[19] Zhize Li, Dmitry Kovalev, Xun Qian, and Peter Richtárik. Acceleration for compressed gradient descent in distributed and federated optimization. In *International Conference on Machine Learning*, pages 5895–5904. PMLR, 2020.

[20] Zhize Li, Hongyan Bao, Xiangliang Zhang, and Peter Richtárik. PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *International Conference on Machine Learning*, 2021.

[21] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pages 3384–3392, 2015.

[22] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[23] Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*, 2019.

[24] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. In *Doklady AN USSR*, volume 269, pages 543–547, 1983.

[25] Yurii Nesterov. *Introductory lectures on convex optimization: a basic course*. Kluwer Academic Publishers, 2004.

[26] Xun Qian, Peter Richtárik, and Tong Zhang. Error compensated distributed sgd can be accelerated. *arXiv preprint arXiv:2010.00091*, 2020.

[27] Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[28] Sebastian U. Stich. Local SGD converges fast and communicates little. In *International Conference on Learning Representations*, 2019.

[29] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient sparsification for communication-efficient distributed optimization. In *Advances in Neural Information Processing Systems*, pages 1306–1316, 2018.

[30] Tian Ye, Peijun Xiao, and Ruoyu Sun. DEED: A general quantization scheme for communication efficiency in bits. *arXiv preprint arXiv:2006.11401*, 2020.

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] We prove the theoretical results for the general convex setting and we leave the nonconvex setting for future work.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This is a pure theoretical work, we did not see the potential negative societal impacts.
   (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] All theorems, lemmas, and corollaries explicitly state all assumptions required by them.
   (b) Did you include complete proofs of all theoretical results? [Yes] Full detailed proofs are provided in the appendix.
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)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (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)? [N/A]
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? [N/A]
   (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? [N/A]
   (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]