Smoothness-Aware Quantization Techniques

Bokun Wang[1] **Mher Safaryan**[1] **Peter Richtárik**[1]

Abstract

Distributed machine learning has become an indispensable tool for training large supervised machine learning models. To address the high communication costs of distributed training, a large body of work has been devoted in recent years to the design of various compression strategies, such as sparsification and quantization, and optimization algorithms capable of using them. Recently, Safaryan et al. (2021) pioneered a dramatically different compression design approach: they first use the local training data to form local *smoothness matrices*, and then propose to design a compressor capable of exploiting the smoothness information contained therein. While this novel approach leads to substantial savings in communication, it is limited to sparsification as it crucially depends on the linearity of the compression operator. It is an open problem whether this approach can be useful in the design of other smoothness-aware compression techniques, such as quantization.

In this work, we resolve this problem by extending their smoothness-aware compression strategy to arbitrary unbiased compression operators, which also includes sparsification. Specializing our results to quantization, we observe significant savings in communication complexity compared to standard quantization. In particular, we show theoretically that block quantization with n blocks outperforms single block quantization, leading to a reduction in communication complexity by an $\mathcal{O}(n)$ factor, where n is the number of nodes in the distributed system. Finally, we provide extensive numerical evidence that our smoothness-aware quantization strategies outperform existing quantization schemes as well the aforementioned smoothness-aware sparsification strategies with respect to all relevant success measures: the number of iterations, the total amount of bits communicated, and wall-clock time.

1. Introduction

Training modern machine learning models is typically cast in terms of (regularized) empirical risk minimization problem and requires increasingly more training data to make empirical risk closer to the true risk (Schmidhuber, 2015; Vaswani et al., 2019). This natural requirement makes it harder (and in some scenarios impossible) to collect all data in one place and carry out the training using a single data source. As a result, we reconciled with a flock of datasets disseminated across various compute nodes holding the actual training data (Bekkerman et al., 2011; Vogels et al., 2019). However, such divide-and-conquer approach of handling vast amount of data means that local updates need to be communicated among the nodes (or through some central server orchestrating the process), which often forms the main bottleneck in modern distributed systems (Zhang et al., 2017; Lin et al., 2018). This issue is further exacerbated by the fact that modern highly performing models are typically overparameterized (Brown et al., 2020; Narayanan et al., 2021).

1.1. Distributed training

In general, distributed training can be formalized as the following optimization problem

$$\min_{x\in\mathbb{R}^d} f(x)+R(x), \quad \text{where} \quad f(x) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(x), \quad (1)$$

and where d is the number of parameters of model $x \in \mathbb{R}^d$ to be trained, n is the number of machines/nodes participating in the training, $f_i(x)$ is the loss/risk associated with the data stored on machine $i \in [n] \overset{\text{def}}{=} \{1,2,\ldots,n\}$, $f(x)$ is the empirical loss/risk, and $R(x)$ is a regularizer.

Because of the communication constraints, large body of work has been devoted in recent years to the design of various compression strategies, such as sparsification (Konečný & Richtárik, 2018; Wangni et al., 2018; Alistarh et al.,

[1] KAUST, KSA. Correspondence to: Mher Safaryan <mher.safaryan@gmail.com>.

This work was presented at the International Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction with ICML 2021 (FL-ICML'21). This workshop does not have official proceedings and this paper is non-archival.

2018), quantization (Goodall, 1951; Roberts, 1962; Alistarh et al., 2017), low-rank approximation (Vogels et al., 2019), and optimization algorithms capable of using them, such as Distributed Compressed Gradient Descent (DCGD) (Khirirat et al., 2018), QSGD (Alistarh et al., 2017; Faghri et al., 2020), NUQSGD (Ramezani-Kebrya et al., 2021), DIANA (Mishchenko et al., 2019; Horváth et al., 2019), PowerSGD (Vogels et al., 2019), signSGD (Bernstein et al., 2018; Safaryan & Richtárik, 2021), intSGD (Mishchenko et al., 2021), ADIANA (Li et al., 2020), MARINA (Gorbunov et al., 2021).

1.2. From scalar smoothness to matrix smoothness

Typically, distributed optimization algorithms in the literature that employ compressed communication, including all methods from the aforementioned works, use only shallow smoothness information of the loss function such as scalar L-smoothness (Nesterov, 2004).

Definition 1 (Scalar Smoothness)**.** Differentiable function $\phi : \mathbb{R}^d \to \mathbb{R}$ is called L-smooth if there exists a nonnegative scalar value $L \ge 0$ such that

$$\phi(x) \le \phi(y) + \langle \nabla \phi(y), x - y\rangle + \tfrac{L}{2}\|x-y\|^2, \qquad (2)$$

for all $x, y \in \mathbb{R}^d$.

As pointed out by Safaryan et al. (2021), smoothness constant L reflects small part of the rich smoothness information often easily available through the training data. In their recent work, Safaryan et al. (2021) pioneered a dramatically different compression design approach. First, they propose to use the local training data to form local *smoothness matrices*, which they claim contain much more useful information than standard smoothness constants.

Definition 2 (Matrix Smoothness)**.** Differentiable function $\phi : \mathbb{R}^d \to \mathbb{R}$ is called $\mathbf{L}$-smooth if there exists a symmetric positive semidefinite matrix $\mathbf{L} \succeq \mathbf{0}$ such that

$$\phi(x) \le \phi(y) + \langle \nabla \phi(y), x - y\rangle + \tfrac{1}{2}\|x-y\|^2_{\mathbf{L}}, \qquad (3)$$

for all $x, y \in \mathbb{R}^d$.

Using smoothness matrices $\mathbf{L}_i$ of all local loss functions $f_i(x),\ i \in [n]$, Safaryan et al. (2021) design a compressor capable of exploiting the smoothness information contained within the smoothness matrices. In particular, under certain heterogeneity conditions on the smoothness matrices $\mathbf{L}_i$, their new compressor reduces total communication cost by a factor of $\mathcal{O}(\min(n,d))$.

> *While this novel approach leads to substantial savings in communication, it is limited to random sparsification as it crucially depends on the linearity of the compression operator. It is not clear whether this approach can be useful in the design of other smoothness-aware compression techniques.*

2. Summary of Contributions

Motivated by the above mentioned development, in this work, we made the following contributions.

2.1. Extending matrix-smoothness-aware sparsification to general compression schemes

First, we generalize the smoothness-aware sparsification strategy (Safaryan et al., 2021) to arbitrary unbiased compressors. Instead of sparsification operator, we consider the generic class $\mathbb{B}(\omega)$ of (possibly randomized) unbiased compression operators $\mathcal{C} \colon \mathbb{R}^d \to \mathbb{R}^d$ with bounded variance $\omega \ge 0$, i.e.,

$$\mathbb{E}\left[\mathcal{C}(x)\right] = x, \quad \mathbb{E}\left[\|\mathcal{C}(x) - x\|^2\right] \le \omega\|x\|^2,$$

for all $x \in \mathbb{R}^d$. This class is quite broad including random sparsification and various quantization schemes. To benefit from the matrix smoothness information with general compressor $\mathcal{C}$, we propose the following modification in the communication protocol. If $x \in \mathbb{R}^d$ is the vector to be communicated, instead of applying compressor $\mathcal{C}$ directly to x and sending $\mathcal{C}(x)$, we compress it by $\mathcal{C}(\mathbf{L}^{\dagger 1/2}x)$ and decompress it by multiplying $\mathbf{L}^{1/2}$. Overall, the receiver estimates the original x by $\mathbf{L}^{1/2}\mathcal{C}(\mathbf{L}^{\dagger 1/2}x)$.

2.2. Distributed compressed methods with improved communication complexity

To highlight the appropriateness of our generalization, we redesign two distributed compressed methods—DCGD (Khirirat et al., 2018) and DIANA (Mishchenko et al., 2019)—to effectively utilize both matrix smoothness information and general compression operators leading to new methods, which we call DCGD+ (Algorithm 1) and DIANA+ (Algorithm 2). The key notion we introduce that enables the technical analysis is the following quantity describing interaction between compression operator $\mathcal{C} \in \mathbb{B}^d(\omega)$ and smoothness matrix $\mathbf{L} \succeq \mathbf{0}$:

$$\mathcal{L}(\mathcal{C}, \mathbf{L}) \overset{\text{def}}{=} \inf\left\{\mathcal{L} \ge 0 \colon \mathbb{E}\|\mathcal{C}(x) - x\|^2_{\mathbf{L}} \le \mathcal{L}\|x\|^2,\ \forall x\right\}.$$

This quantity generalizes the one defined in (Safaryan et al., 2021) for sparsification, and provides means for tighter theoretical guarantees (Theorems 1 and 2) and better compression design. Notice that $\mathcal{L}(\mathcal{C}, \mathbf{L}) \le \omega\lambda_{\max}(\mathbf{L})$.

2.3. Block quantization

As we are no longer constrained to sparsification to exploit matrix smoothness, we consider more aggressive quantization schemes to further reduce the communication cost. Our

Table 1. Summary of main theoretical results of this work. Below constants and $\log\frac{1}{\varepsilon}$ factors are hidden, n is the number of nodes, d is the model size, $L_{\max} = \max_i L_i,\ L_i = \lambda_{\max}(\mathbf{L}_i)$, the expected smoothness constant $\mathcal{L}_{\max}$ is defined in (4), the variance of generic compression operator is denoted by ω, parameters ν and ν_1 are defined in (8). Refer the notation table in the Appendix.

Regime	$\nabla f_i(x^*) \equiv 0$	arbitrary $\nabla f_i(x^*)$
Original Methods	**DCGD (Khirirat et al., 2018)**	**DIANA (Mishchenko et al., 2019)**
Iteration Complexity	$\frac{L}{\mu} + \frac{\omega L_{\max}}{n\mu}$	$\omega + \frac{L_{\max}}{\mu} + \frac{\omega L_{\max}}{n\mu}$
Communication Complexity Standard Quantization ($\omega = \mathcal{O}(n)$)	$d\frac{L_{\max}}{\mu}$	$nd + d\frac{L_{\max}}{\mu}$
Redesigned Methods	**DCGD+ (Algorithm 1) with general compression**	**DIANA+ (Algorithm 2) with general compression**
Iteration Complexity	$\frac{L}{\mu} + \frac{\mathcal{L}_{\max}}{n\mu}$	$\omega_{\max} + \frac{L}{\mu} + \frac{\mathcal{L}_{\max}}{n\mu}$
Communication Complexity Block Quantization ($n = \mathcal{O}(\sqrt{d})$)	$\frac{d}{n}\frac{L_{\max}}{\mu}$ (if $\nu,\ \nu_1$ are $\mathcal{O}(1)$)	$nd + \frac{d}{\sqrt{nd}}\frac{L_{\max}}{\mu}$ (if $\nu,\ \nu_1$ are $\mathcal{O}(1)$)
Communication Complexity Quantization with varying steps	$\frac{d}{n}\frac{L_{\max}}{\mu} + \frac{d}{d}\frac{L_{\max}}{\mu}$ (if $\nu,\ \nu_1$ are $\mathcal{O}(1)$)	$nd + \frac{d}{n}\frac{L_{\max}}{\mu} + \frac{d}{d}\frac{L_{\max}}{\mu}$ (if $\nu,\ \nu_1$ are $\mathcal{O}(1)$)
Theorems	1, 3, 5	2, 4, 6
Speedup factor (up to)	$\min(n, d)$	$\min(n, d)$

first extension of standard quantization (Alistarh et al., 2017) is *block quantization*, where each block is allowed to have a separate quantization parameter. Notably, we show theoretically that our block quantization with n blocks outperforms single block quantization and saves in communication by a factor of $\mathcal{O}(n)$ for both DCGD+ (Theorem 3) and DIANA+ (Theorem 4) when $n = \mathcal{O}(\sqrt{d})$.

2.4. Quantization with varying steps

In our second extension of standard quantization, we go even further and allow all coordinates to have their own quantization steps. This extension turns out to be more efficient in practice than block quantization and provides savings in communication cost by a factor of $\mathcal{O}(\min(n, d))$ for both DCGD+ (Theorem 5) and DIANA+ (Theorem 6).

2.5. Experiments

Finally, we perform extensive numerical experiments using LibSVM data (Chang & Lin, 2011) and provide clear numerical evidence that the proposed smoothness-aware quantization strategies outperform existing quantization schemes as well the aforementioned smoothness-aware sparsification strategies with respect to all relevant success measures: the number of iterations, the total amount of bits communicated, and wall-clock time (see Section 6 and the Appendix).

3. Smoothness-Aware Distributed Methods with General Compressors

In this section we extend methods DCGD+ and DIANA+ of (Safaryan et al., 2021) to handle arbitrary unbiased compression operators. We consider the problem (1) with matrix smoothness assumption for all local losses $f_i(x)$ and with strong convexity of loss function $f(x)$.

Assumption 1 (Matrix smoothness)**.** The functions $f_i\colon \mathbb{R}^d \to \mathbb{R}$ are differentiable, convex, lower bounded and $\mathbf{L}_i$-smooth. Besides, f is $\mathbf{L}$-smooth with the scalar smoothness constant $L \overset{\text{def}}{=} \lambda_{\max}(\mathbf{L})$.

Note that lower boundedness of $f_i(x)$ is not needed once $\mathbf{L}_i \succ 0$ is invertible. This part of the assumption is not a restriction in applications as all loss function are lower bounded.

Assumption 2 (μ-convexity)**.** The function $f\colon \mathbb{R}^d \to \mathbb{R}$ is μ-convex for some $\mu > 0$, i.e.,

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \tfrac{\mu}{2}\|x - y\|^2,$$

for all $x, y \in \mathbb{R}^d$.

3.1. DCGD+ with arbitrary unbiased compression

In our version of DCGD+, each node $i \in [n]$ is allowed to control its own compression operator $\mathcal{C}_i \in \mathbb{B}(\omega)$ independent of other nodes. Denote

$$\mathcal{L}_{\max} \overset{\text{def}}{=} \max\nolimits_{1 \leq i \leq n} \mathcal{L}_i, \quad \text{where} \quad \mathcal{L}_i \overset{\text{def}}{=} \mathcal{L}(\mathcal{C}_i, \mathbf{L}_i). \quad (4)$$

Furthermore, as the compressor $\mathcal{C}_i$ can be random, denote by $\mathcal{C}_i^k$ a copy of $\mathcal{C}_i$ generated at iteration k.

Algorithm 1 DCGD+ WITH GENERAL COMPRESSION

```
Input: Initial point $x^0 \in \mathbb{R}^d$, step size $\gamma > 0$, compression operators $\{\mathcal{C}_1^k, \dots, \mathcal{C}_n^k\}$
on server
    send $x^k$ to all nodes
    get $\mathcal{C}_i^k(\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^k))$ from all nodes $i \in [n]$
    $g^k = \frac{1}{n}\sum_{i=1}^n \mathbf{L}_i^{1/2}\mathcal{C}_i^k(\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^k))$
    update the model to $x^{k+1} = \mathrm{prox}_{\gamma R}(x^k - \gamma g^k)$
```

Similar to the standard DCGD method, convergence of DCGD+ is linear up to some oscillation neighborhood. However, for overparametrized models this neighborhood vanishes and the method converges linearly to the exact solution.

Theorem 1. *Let Assumptions 1 and 2 hold and assume that each node $i \in [n]$ generates its own copy of compression operator $\mathcal{C}_i^k \in \mathbb{B}^d(\omega_i)$ independently from others. Then, for the step-size $0 < \gamma \le \frac{1}{L + \frac{2}{n}\mathcal{L}_{\max}}$, the iterates $\{x^k\}$ of DCGD+ (Algorithm 1) satisfy*

$$\mathbb{E}\left[\|x^k - x^*\|^2\right] \le (1-\gamma\mu)^k \|x^0 - x^*\|^2 + \frac{2\gamma\sigma_+^*}{\mu n}, \tag{5}$$

where $\sigma_+^* \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n \mathcal{L}_i \|\nabla f_i(x^*)\|^2_{\mathbf{L}_i^\dagger}$. In particular, if the model is overparameterized (i.e., $\nabla f_i(x^*) = 0$ for all $i \in [n]$), then DCGD+ converges linearly with iteration complexity

$$\mathcal{O}\left(\left(\frac{L}{\mu} + \frac{\mathcal{L}_{\max}}{n\mu}\right)\log\frac{1}{\varepsilon}\right). \tag{6}$$

We show later that the linear rate (6) of DCGD+ can be much better than one of DCGD. However, the size of the neighborhood of DCGD+ might be bigger than of DCGD. In case of standard (scalar) smoothness (i.e. $\mathbf{L}_i = L_i\mathbf{I}$) the size of the neighborhood would be $\sigma^* \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n \omega_i\|\nabla f_i(x^*)\|^2$, which might be smaller than σ_+^*. Even though we have $\mathcal{L}_i \le \omega_i\lambda_{\max}(\mathbf{L}_i)$ from the definition of $\mathcal{L}_i$, it does not imply $\mathcal{L}_i\mathbf{L}_i^\dagger \preceq \omega_i\mathbf{I}$. Thus, with matrix-smoothness-aware compression we ensure faster linear convergence at the cost of a possibly larger oscillation radius. This is not an issue for modern overparameterized models, which can interpolate the whole training data with zero loss. Moreover, next we present an algorithmic solution to remove the neighborhood using the DIANA method.

3.2. DIANA+ with arbitrary unbiased compression

The mechanism allowing to remove the neighborhood in DIANA+ is based on the DIANA method, which was initially introduced for ternary quantization by Mishchenko et al. (2019), and then extended to arbitrary unbiased compression operators by Horváth et al. (2019). The high level idea is to learn the local optimal gradients $\nabla f_i(x^*)$ by estimates u_i^k for all nodes $i \in [n]$ in a communication efficient manner. Nodes use these estimates u_i^k to progressively construct better local gradient estimates g_i^k reducing the variance induced from the compression.

Algorithm 2 DIANA+ WITH GENERAL COMPRESSION

```
Input: Initial point $x^0 \in \mathbb{R}^d$, initial shifts $u_i^0 \in \mathrm{range}(\mathbf{L}_i)$ and $u^0 \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n u_i^0$, step size parameters $\gamma > 0$ and $\alpha > 0$, compression operators $\{\mathcal{C}_1^k, \dots, \mathcal{C}_n^k\}$
for each node $i = 1, \dots, n$ in parallel do
    get $x^k$ from the server and compute $\nabla f_i(x^k)$
    send $\Delta_i^k = \mathcal{C}_i^k(\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x^k) - u_i^k))$ to the server
    update local gradient and shift
    $\overline{\Delta}_i^k = \mathbf{L}_i^{1/2}\Delta_i^k,\ g_i^k = u_i^k + \overline{\Delta}_i^k, u_i^{k+1} = u_i^k + \alpha\overline{\Delta}_i^k$
end for
on server
    get sparse updates $\Delta_i^k$ from all nodes $i \in [n]$
    $\overline{\Delta}^k = \frac{1}{n}\sum_{i=1}^n \overline{\Delta}_i^k,\ g^k = \overline{\Delta}^k + u^k$
    update the global model $x^{k+1} = \mathrm{prox}_{\gamma R}(x^k - \gamma g^k)$
    update the global shift $u^{k+1} = u^k + \alpha\overline{\Delta}^k$
```

We prove in the Appendix that both iterates x^k and all local gradient estimates u_i^k converge linearly to the exact solution x^* and $\nabla f_i(x^*)$ respectively.

Theorem 2. *Let Assumptions 1 and 2 hold and assume that each node $i \in [n]$ generates its own copy of compression operator $\mathcal{C}_i^k \in \mathbb{B}^d(\omega_i)$ independently from others. Then, for the step-size $\gamma = \frac{1}{L + \frac{6}{n}\mathcal{L}_{\max}}$, DIANA+ (Algorithm 2) guarantees $\mathbb{E}\left[\|x^k - x^*\|^2\right] \le \varepsilon$ after*

$$\mathcal{O}\left(\left(\omega_{\max} + \frac{L}{\mu} + \frac{\mathcal{L}_{\max}}{n\mu}\right)\log\frac{1}{\varepsilon}\right) \tag{7}$$

iterations, where $\omega_{\max} = \max_{1 \le i \le n}\omega_i$.

Notice that the cost of removing the neighborhood is the extra $\mathcal{O}(\omega_{\max}\log\frac{1}{\varepsilon})$ iterations, which is negligible in the overall complexity (7) above. Another interesting observation is the second order flavor of the gradient learning technique employed by DIANA+. Let, for concreteness, matrices $\mathbf{L}_i$ be invertible and $\mathcal{C}_i^k(-x) = -\mathcal{C}_i^k(x)$ for all $x \in \mathbb{R}^d$ (both random sparsification and quantization satisfy this). Typically, the learning procedure of the original DIANA method, $u_i^{k+1} = u_i^k - \alpha\mathcal{C}_i^k(u_i^k - \nabla f_i(x^k))$, can be interpreted as a single step of CGD applied to the problem of minimizing the convex quadratic function $\varphi_i^k(u) \overset{\text{def}}{=} \frac{1}{2}\left\|u - \nabla f_i(x^k)\right\|^2$, which changes in each iteration because the gradient changes. In contrast, we observe that the learning mechanism of DIANA+ can be interpreted as a single step of a (dumped) Newton's method with compressed gradients and with the true Hessian. Indeed, fix the iteration counter k and denote

$$\varphi_i^k(u) \overset{\text{def}}{=} \tfrac{1}{2}\left\|u - \nabla f_i(x^k)\right\|^2_{\mathbf{L}_i^{-1/2}}.$$

Then, the update rule of shifts u_i^k in DIANA+ can be rewritten as

$$\begin{aligned} u_i^{k+1} &= u_i^k - \alpha \mathbf{L}_i^{1/2}\mathcal{C}_i^k(\mathbf{L}_i^{-1/2}(u_i^k - \nabla f_i(x^k))) \\ &= u_i^k - \alpha\left[\nabla^2 \varphi_i^k(u_i^k)\right]^{-1}\mathcal{C}_i^k(\nabla\varphi_i^k(u_i^k)). \end{aligned}$$

This might serve as an extra explanation on why incorporating smoothness matrices properly can improve the performance of first order methods with communication compression.

3.3. Baselines for the original methods

To make the theoretical comparison against DCGD and DIANA more transparent, we fix the following baselines using the standard quantization scheme.

• **Baseline for DCGD.** Based on the iteration complexity $\widetilde{\mathcal{O}}(\frac{L}{\mu} + \frac{\omega L_{\max}}{n\mu})$ of DCGD (in case $\nabla f_i(x^*) = 0$ for all $i \in [n]$) the optimal level of compression variance $\omega = \mathcal{O}(n)$ results in $\widetilde{\mathcal{O}}(\frac{L_{\max}}{\mu})$ iterations complexity. From the estimate of quantization variance $\omega = \min\left(\frac{d}{s^2}, \frac{\sqrt{d}}{s}\right)$ we conclude that $s = \mathcal{O}(\frac{\sqrt{d}}{n})$ number of levels should be used. Finally, with this choice of s, each node communicates $\mathcal{O}(s^2 + s\sqrt{d}) = \mathcal{O}(\frac{d}{n})$ amount of bits. Thus, total communication complexity (i.e. how many bits flows through the central server) of DCGD is $\widetilde{\mathcal{O}}(\frac{dL_{\max}}{\mu})$.

• **Baseline for DIANA.** Based on the iteration complexity $\widetilde{\mathcal{O}}(\omega + \frac{L_{\max}}{\mu} + \frac{\omega L_{\max}}{n\mu})$ of DIANA method, we fix the same level of compression with $\omega = \mathcal{O}(n)$. With a similar argument, this leads us to $\widetilde{\mathcal{O}}(n + \frac{L_{\max}}{\mu})$ iteration complexity and $\mathcal{O}(\frac{d}{n})$ bits of communication per node in each iteration. Whence, total communication complexity becomes $\widetilde{\mathcal{O}}(dn + \frac{dL_{\max}}{\mu})$.

To compare the proposed methods with these baselines and highlight improvement factors, define parameters ν and ν_1 describing local smoothness matrices $\mathbf{L}_i$ as follows

$$\nu \overset{\text{def}}{=} \frac{\sum_{i=1}^n L_i}{\max_{i\in[n]} L_i}, \quad \nu_1 \overset{\text{def}}{=} \max_{i\in[n]} \frac{\sum_{j=1}^d \mathbf{L}_{i;j}}{\max_{j\in[d]} \mathbf{L}_{i;j}}, \tag{8}$$

where $L_i = \lambda_{\max}(\mathbf{L}_i)$, $L_{\max} \overset{\text{def}}{=} \max_{1\le i\le n} L_i$ and $\mathbf{L}_{i;j}$ is the jth diagonal element of matrix $\mathbf{L}_i$. Parameters $\nu \in [1, n]$ and $\nu_1 \in [1, d]$ describe the level of heterogeneity over the nodes and coordinates respectively. If $\mathbf{L}_i$ matrices coincide, then $\nu = n$ and $\nu_1 = d$. On the other extreme, when the values of $\mathbf{L}_i$ are extremely non-uniform, we have $\nu \ll n$ and $\nu_1 \ll d$.

Notice that the quantity $\frac{\mathcal{L}_{\max}}{\mu n}$ in (6) and the quantity $\omega_{\max} + \frac{\mathcal{L}_{\max}}{\mu n}$ in (7) depend on compression operators $\mathcal{C}_i^k$ applied by the nodes. For the rest of the paper we are going to minimize these quantities with respect to the choice of $\mathcal{C}_i^k$ in such a way to minimize total communication complexity of the proposed distributed methods. We specialize compressors $\mathcal{C}_i$ to two different extensions of standard quantization and optimize with respect to compression parameters.

4. Block Quantization

We now present our first extension to standard quantization in order to properly capture the matrix smoothness information. Instead of having a single quantization parameter (e.g. number of levels) for all coordinates, here we divide the space $\mathbb{R}^d$ into $B \in \{1, 2, \dots, d\}$ blocks as $\mathbb{R}^d = \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \times \cdots \times \mathbb{R}^{d_B}$ and for each subspace $\mathbb{R}^{d_l}$, $l \in [B]$ we apply standard quantization independently from other blocks with different number of levels s_l. Thus, for any $l \in [B]$ we allocate one parameter s_l for l^{th} block of $x \in \mathbb{R}^d$. Hence quantization is applied block-wise: for each block we send the norm $\|x^l\|$ of the block $x^l \in \mathbb{R}^{d_l}$ and all entries within this block are quantized with levels $\{0, \frac{1}{s_l}, \frac{2}{s_l}, \dots, 1\}$. In the special case of $B = 1$, we get the standard quantization of Alistarh et al. (2017).

To get rid of the constraints on s_l to be integers, instead of working with the number of levels s_l, we introduce the size of the quantization step $h_l = \frac{1}{s_l}$ and allow them to take any positive values (even bigger than 1). Thus, for each block $l \in [B]$ we quantize with respect to levels $\{0, h_l, 2h_l, \dots\}$.

Definition 3 (Block Quantization)**.** For a given number of blocks $B \in [d]$ and fixed quantization steps $h = (h_1, \dots, h_B)$, define block-wise quantization operator $\mathcal{Q}_h^B \colon \mathbb{R}^d \to \mathbb{R}^d$ as follows:

$$\left[\mathcal{Q}_h^B(x)\right]_t \overset{\text{def}}{=} \|x^l\| \cdot \operatorname{sign}(x_t) \cdot \xi_l\left(\tfrac{|x_t|}{\|x^l\|}\right),$$

where $x \in \mathbb{R}^d$, $t = (l-1)B + j$, $j \in [d_l]$, $l \in [B]$ and $\xi_l(v)$ for $v \ge 0$ is defined via the quantization levels $\{0, h_l, 2h_l, \dots\}$ as follows: if $kh_l \le v < (k+1)h_l$ for some $k \in \{0, 1, 2, \dots\}$, then

$$\xi_l(v) \overset{\text{def}}{=} \begin{cases} kh_l & \text{with prob.} \quad k + 1 - \frac{v}{h_l} \\ (k+1)h_l & \text{with prob.} \quad \frac{v}{h_l} - k \end{cases}. \tag{9}$$

Note that $\mathcal{Q}_h^B$ is an unbiased compression operator as $\mathbb{E}\left[\xi_j(v)\right] = v$ for any $v \ge 0$. To communicate a vector of the form $\mathcal{Q}_h^B(x)$, we encode each block $\left[\mathcal{Q}_h^B(x)\right]^l \in \mathbb{R}^{d_l}$ using Elias ω-coding as in the standard quantization scheme (Alistarh et al., 2017). Hence, for each block $l \in [B]$ we need to send $\widetilde{\mathcal{O}}(\frac{1}{h_l^2} + \frac{\sqrt{d_l}}{h_l})$ bits and one floating point number for $\|x^l\|$. Overall, the number of encoding bits for $\mathcal{Q}_h^B(x)$ (up to constant and $\log$ factors) can be given by $\sum_{l=1}^B(\frac{1}{h_l^2}+\frac{\sqrt{d_l}}{h_l})+B$. As for the compression noise, we prove in the Appendix the following upper bound for $\mathcal{L}(\mathcal{Q}_h^B,\mathbf{L})$:

$$\mathcal{L}(\mathcal{Q}_h^B,\mathbf{L}) \le \max\nolimits_{1\le l\le B} h_l \| \mathbf{Diag}(\mathbf{L}^{ll})\|, \qquad (10)$$

where $\mathbf{L}^{ll}$ is the l^{th} diagonal block matrix of $\mathbf{L}$ with sizes $d_l \times d_l$. Next, we are going to minimize communication complexity of DCGD+ and DIANA+ by optimizing parameters of block quantization.

4.1. DCGD+ with block quantization

We fix the number of blocks $B \in [d]$ for all nodes $i \in [n]$ and allow each node to apply different block quantization operator $\mathcal{Q}_{h_i}^B$ with quantization steps $h_i = (h_{i,1},\dots,h_{i,B})$. To minimize communication complexity of DCGD+, we need to minimize $\mathcal{L}_{\max}$ subject to the communication constraint mentioned above. Since $\mathcal{L}_{\max} = \max_{i\in[n]}\mathcal{L}(\mathcal{C}_i,\mathbf{L}_i)$, each node $i\in[n]$ can minimize the impact of its own compression by minimizing $\mathcal{L}(\mathcal{C}_i,\mathbf{L}_i)$ based on local smoothness matrix $\mathbf{L}_i$. This leads to the following optimization problem for finding optimal values of h_i for each node $i\in[n]$:

$$\begin{aligned} \min_{h\in\mathbb{R}^B} \quad & \max\nolimits_{1\le l\le B} h_l \| \mathbf{Diag}(\mathbf{L}_i^{ll})\| \\ \text{s.t.} \quad & \textstyle\sum_{l=1}^B \left(\frac{1}{h_l^2}+\frac{\sqrt{d_l}}{h_l}\right)+B=\beta \\ & h_l>0,\ l\in[B] \end{aligned} \qquad (11)$$

The solution to this problem is given by

$$h_{i,l} = \frac{\delta_{i,B}}{\|\mathbf{Diag}(\mathbf{L}_i^{ll})\|}, \qquad (12)$$

where $\delta_{i,B}\ge 0$ is uniquely determined by the constraint equality of (11) as the only positive solution of

$$\delta_{i,B}^2 - \delta_{i,B}\frac{dT_{i,B}}{\beta-B} - \frac{dT_{1,B}^2}{\beta-B} = 0,$$

which implies $\delta_{i,B} = \frac{dT_{i,B}}{2(\beta-B)} + \sqrt{\frac{d^2T_{i,B}^2}{4(\beta-B)^2}+\frac{dT_{i,1}^2}{\beta-B}} \le \frac{d}{\beta-B}T_{i,B} + \sqrt{\frac{d}{\beta-B}}T_{i,1}$, where $T_{i,B} \overset{\text{def}}{=} \frac{1}{d}\sum_{l=1}^B\sqrt{d_l}\|\mathbf{Diag}(\mathbf{L}_i^{ll})\|$. If this solution of quantization steps h_i is used by all nodes $i\in[n]$, then we show reduction in communication complexity by a factor of $\mathcal{O}(n)$ compared to standard quantization.

Theorem 3. *Assume $n=\mathcal{O}(\sqrt{d})$ and both ν,ν_1 are $\mathcal{O}(1)$. Then DCGD+ using block quantization with $B=n$ blocks, $d_l=\mathcal{O}(d/n)$ block sizes for all $l\in[n]$ and quantization steps (12) with $\beta=\mathcal{O}(d/n)$ reduces overall communication complexity by a factor of $\mathcal{O}(n)$ compared to DCGD using $B=1$ single block quantization. Formally, to guarantee $\varepsilon>0$ accuracy, the communication complexity of DCGD+ is*

$$\mathcal{O}\left(\frac{d}{n}\frac{L_{\max}}{\mu}\log\frac{1}{\varepsilon}\right),$$

which is $\mathcal{O}(n)$ times smaller over DCGD.

4.2. DIANA+ with block quantization

For the rate (7) of DIANA+, we need to optimize $\omega_{\max}+\frac{\mathcal{L}_{\max}}{n\mu}$ part of the complexity under the same communication constraint used in (11). Since

$$\max_{i\in[n]}\left(\omega_i+\frac{\mathcal{L}_i}{n\mu}\right) \le \omega_{\max}+\frac{\mathcal{L}_{\max}}{n\mu} \le 2\max_{i\in[n]}\left(\omega_i+\frac{\mathcal{L}_i}{n\mu}\right), \qquad (13)$$

we can decompose the problem into subproblems for each node i to optimize $\omega_i+\frac{\mathcal{L}_i}{n\mu}$ with respect to its own quantization parameters h_i. Analogously, this leads to the following optimization problem for finding optimal values of h_i for each node $i\in[n]$:

$$\begin{aligned} \min_{h\in\mathbb{R}^B} \quad & \max\nolimits_{1\le l\le B} h_l\left(\sqrt{d_l}+\tfrac{1}{\mu n}\|\mathbf{Diag}(\mathbf{L}_i^{ll})\|\right) \\ \text{s.t.} \quad & \textstyle\sum_{l=1}^B\left(\frac{1}{h_l^2}+\frac{\sqrt{d_l}}{h_l}\right)+B=\beta \\ & h_l>0,\ l\in[B] \end{aligned} \qquad (14)$$

which can be solved with a similar argument as done for (11). Details are deferred to the Appendix.

Theorem 4. *Assume $n=\mathcal{O}(\sqrt{d})$ and both ν,ν_1 are $\mathcal{O}(1)$. Then DIANA+ using block quantization with $B=n$ blocks, $d_l=\mathcal{O}(d/n)$ block sizes for all $l\in[n]$ and $h_{i,l}$ quantization steps (solution to (14)) with $\beta=\mathcal{O}(d/n)$ reduces overall communication complexity by a factor of $\mathcal{O}(n)$ compared to DIANA using $B=1$ single block quantization. Formally, to guarantee $\varepsilon>0$ accuracy, the communication complexity of DIANA+ is*

$$\mathcal{O}((nd+\sqrt{\tfrac{d}{n}}\tfrac{L_{\max}}{\mu})\log\tfrac{1}{\varepsilon}),$$

which (ignoring n summand in the complexity) is $\mathcal{O}(n)$ times smaller over DIANA.

5. Quantization with Varying Steps

Our second extension of standard quantization scheme is to allow different quantization steps for all coordinates $\{1,2,\dots,d\}$. In other words, for each coordinate $j\in[d]$ we quantize with respect to levels $\{0,h_j,2h_j,\dots\}$. The standard quantization (Alistarh et al., 2017) is the special case when $h_j=\frac{1}{s}$ for all $j\in[d]$, where s is the number of quantization levels.

Definition 4 (Quantization with varying steps)**.** For fixed quantization steps $h=(h_1,\dots,h_d)^\top\in\mathbb{R}^d$, define quantization operator $\mathcal{Q}_h\colon\mathbb{R}^d\to\mathbb{R}^d$ as follows:

$$[\mathcal{Q}_h(x)]_j = \|x\|\cdot\operatorname{sign}(x_j)\cdot\xi_j\left(\frac{|x_j|}{\|x\|}\right),$$

where $x \in \mathbb{R}^d,\ j = 1, 2, \dots, d$ and ξ_j is defined via the quantization levels $\{0, h_j, 2h_j, \dots\}$ as in (9).

Note that compression operator $\mathcal{Q}_h$ is unbiased as $\mathbb{E}\left[\xi_j(v)\right] = v$ for any $v \geq 0$. To understand how the number of encoding bits of $\mathcal{Q}_h(x)$ depends on h exactly seems challenging, since it depends on the actual encoding scheme (i.e. binary representation of compressed information). Besides, even if we fix binary mapping, the closed form expression of total amount of bits is complicated enough to be utilized in the further analysis. We provide theoretical arguments and clear numerical evidence that $\|h^{-1}\| = \sqrt{\sum_{j=1}^d h_j^{-2}}$ is a reasonable proxy for the number of encoding bits for compressor $\mathcal{Q}_h$.

Assumption 3. For any input vector $x \in \mathbb{R}^d$ and quantization steps $h \in \mathbb{R}^d$, compressed vector $\mathcal{Q}_h(x)$ can be encoded with $\mathcal{O}(\|h^{-1}\|)$ number of bits.

First, consider the special case when all quantization steps are the same, i.e. $h_j = \frac{1}{s}$. Then $\|h^{-1}\| = s\sqrt{d}$ recovers the dominant part (provided $s = \mathcal{O}(\sqrt{d})$) in $\widetilde{\mathcal{O}}(s^2 + s\sqrt{d})$ showing total amount of bits for standard quantization scheme.

Second, in the Appendix we present an encoding scheme which (up to constant and $\log d$ factors) requires $\mathbb{E}\left[\psi(\|\hat{x}\|_0)\right] + \|h^{-1}\|$ number of bits in expectation to communicate $\hat{x} = \mathcal{Q}_h(x)$, where $\psi(\tau) \overset{\text{def}}{=} dH_2(\tau/d) + \tau \leq d\log 3$, if $\tau \in [0, d]$ and H_2 is the binary entropy function. Note that, based on the definition (9), increasing quantization steps h_j forces more sparsity in $\hat{x}$ and hence reduces $\|\hat{x}\|_0$. Thus, $\|\hat{x}\|_0$ and hence $\psi(\|\hat{x}\|_0)$ (notice that $\psi(0) = 0$) are proportional to $\|h^{-1}\|$. Furthermore, we present a numerical experiment which shows that the number of encoding bits of $\mathcal{Q}_h(x)$ and $\|h^{-1}\|$ are positively correlated.

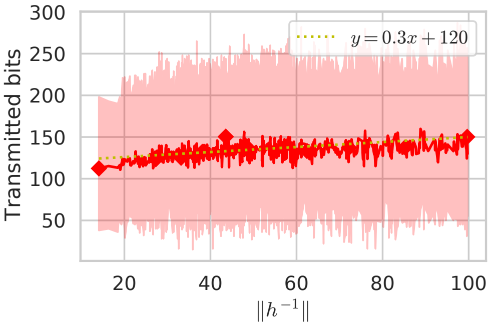

Figure 1. Experiment to verify the Assumption 3. We randomly generate 1000 quantization step vectors $h \in \mathbb{R}^{50}$, each component of h is $h_j = |\tilde{h}_j|$ and $\tilde{h}_j$ is independently sampled from $\mathcal{N}(0, 1)$. For each h, we randomly generate multiple sparse vectors to quantize x, which is sampled from Poisson distribution with $\lambda = \{1, 10, 100\}$ and density $\{0.25, 0.5, 0.75, 1.0\}$.

Hence, in the further analysis, we fix the number of encoding bits of $\mathcal{Q}_h(x)$ by the constraint $\|h^{-1}\| = \beta$ for some parameter $\beta > 0$. As for the variance induced by the compression operator $\mathcal{Q}_h$, we prove the following upper bound for $\mathcal{L}(\mathcal{Q}_h, \mathbf{L})$:

$$\mathcal{L}(\mathcal{Q}_h, \mathbf{L}) \leq \|\,\mathbf{Diag}(\mathbf{L})h\|. \tag{15}$$

5.1. DCGD+ with varying quantization steps

Now, we optimize the rate (6) of DCGD+ with respect to quantization steps $h_i = (h_{i;1}, h_{i;2}, \dots, h_{i;d})$ of compressor $\mathcal{Q}_{h_i}$ controlled by i^{th} node for all $i \in [n]$. The term in (6) affected by the compression is $\mathcal{L}_{\max} = \max_{i\in[n]} \mathcal{L}(\mathcal{C}_i, \mathbf{L}_i)$, which implies that each node $i \in [n]$ can minimize the impact of its own compression by minimizing $\mathcal{L}(\mathcal{C}_i, \mathbf{L}_i)$ based on local smoothness matrix $\mathbf{L}_i$. Based on the upper bound (15) and communication constraint given by $\|h^{-1}\| = \beta$ for some $\beta > 0$, we get the following optimization problem to choose the optimal quantization parameters h_i for node $i \in [n]$:

$$\begin{aligned} \min_{h\in\mathbb{R}^d} \quad & \|\,\mathbf{Diag}(\mathbf{L}_i)h\| \\ \text{s.t.} \quad & \|h^{-1}\| = \beta \\ & h_j > 0,\ j \in [d] \end{aligned} \tag{16}$$

This problem has the following closed form solution due to KKT conditions (see Appendix):

$$h_{i;j} = \frac{1}{\beta}\sqrt{\frac{\sum_{t=1}^d \mathbf{L}_{i;t}}{\mathbf{L}_{i;j}}}, \quad i \in [n],\ j \in [d]. \tag{17}$$

With this choice of quantization steps we save $\mathcal{O}(\min(n, d))$ times in communication.

Theorem 5. *Assume both ν, ν_1 are $\mathcal{O}(1)$ and $\beta = \mathcal{O}(d/n)$. Then DCGD+ using quantization with varying steps (26) for all $i \in [n]$ reduces overall communication complexity by a factor of $\mathcal{O}(\min(n, d))$ compared to the baseline of DCGD. Formally, the iteration complexity (6) can be upper bounded as*

$$\frac{L}{\mu} + \frac{\mathcal{L}_{\max}}{n\mu} \leq \frac{\nu}{n}\frac{L_{\max}}{\mu} + \frac{\nu_1}{\beta}\frac{L_{\max}}{n\mu} = \mathcal{O}\left(\frac{1}{n}\frac{L_{\max}}{\mu} + \frac{1}{d}\frac{L_{\max}}{\mu}\right),$$

which is $\min(n, d)$ times smaller than the one for DCGD. As both methods communicate $\mathcal{O}(d/n)$ bits per node per iteration, we get $\min(n, d)$ times savings in communication complexity.

5.2. DIANA+ with varying quantization steps

Based on (13), each node $i \in [n]$ optimizes $\omega_i + \frac{\mathcal{L}_i}{n\mu}$ with respect to its quantization parameters h_i, which is equivalent to the problem

$$\begin{aligned} \min_{h\in\mathbb{R}^d} \quad & \textstyle\sum_{j=1}^d \left(1 + A_{ij}^2\right) h_j^2 \\ \text{s.t.} \quad & \|h^{-1}\| = \beta \\ & h_j > 0,\ j \in [d] \end{aligned} \tag{18}$$

where $A_{ij} \overset{\text{def}}{=} \frac{\mathbf{L}_{i;j}}{n\mu}$. Due to the KKT conditions (see Appendix), we get the following solution

$$h_{i;j} = \frac{1}{\beta}\sqrt{\frac{\sum_{t=1}^{d}\sqrt{1+A_{it}^2}}{\sqrt{1+A_{ij}^2}}}. \tag{19}$$

With this choice of quantization steps we save $\mathcal{O}(\min(n,d))$ times in communication.

Theorem 6. *Assume both ν, ν_1 are $\mathcal{O}(1)$ and $\beta = \mathcal{O}(d/n)$. Then DIANA+ using quantization with varying steps (29) for all $i \in [n]$ reduces overall communication complexity by a factor of $\mathcal{O}(\min(n,d))$ compared to the baseline of DIANA. Formally, the iteration complexity (7) can be upper bounded as*

$$\begin{aligned}\omega_{\max} + \frac{L}{\mu} + \frac{\mathcal{L}_{\max}}{n\mu} \quad &\leq \quad \frac{\sqrt{2}d}{\beta} + \frac{\nu}{n}\frac{L_{\max}}{\mu} + \frac{\sqrt{2}\nu_1}{\beta n}\frac{L_{\max}}{\mu} \\ &= \quad \mathcal{O}\left(n + \frac{1}{n}\frac{L_{\max}}{\mu} + \frac{1}{d}\frac{L_{\max}}{\mu}\right),\end{aligned}$$

which is $\min(n,d)$ times smaller than the one for DIANA (ignoring negligible term n).

6. Experiments

In this section we present two key experiments. Additional experiments can be found in the Appendix.

6.1. Setup

We run the experiments with several datasets listed in Table 2 from the LibSVM repository (Chang & Lin, 2011) on the ℓ_2-regularized logistic regression problem described below:

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

$$f_i(x) = \frac{1}{m}\sum_{t=1}^{m}\log(1+\exp(-b_{i,t}\mathbf{A}_{i,t}^\top x)) + \frac{\lambda}{2}\|x\|^2,$$

where $x \in \mathbb{R}^d$, $\mathbf{A}_{i,l} \in \mathbb{R}^d$, $b_{i,l} \in \{-1,1\}$ are the feature and label of l-th data point on the i-th worker, where the features of each $\mathbf{A}_{i,l}$ are rescaled into $[-1,1]$. The data points are randomly shuffled before allocating to local workers. The experiments are performed on a workstation with Intel(R) Xeon(R) Gold 6246 CPU @ 3.30GHz cores. The `gather` and `broadcast` operations for the communications between master and workers are implemented based on the MPI4PY library (Dalcín et al., 2005) and each CPU core is treated as a local worker. We set $\lambda = 10^{-3}$ for all datasets. For each dataset, we run each algorithm multiples times with 5 random seeds for each worker.

6.2. Comparison to standard quantization techniques

In our first experiment, we compare smoothness-aware DCGD+ and DIANA+ methods with our varying-step quantization technique (`quant+`) to the original DCGD (Khirirat et al., 2018) and DIANA (Mishchenko et al., 2019) methods with the standard quantization technique (`quant`) of Alistarh et al. (2017). Figure 2 demonstrates that DCGD+/DIANA+ with `quant+` lead to significant improvement in both transmitted megabytes and wall-clock time. An ablation study to disentangle the contributions of exploiting the smoothness matrix and utilizing varying number of levels can be found in Appendix B.

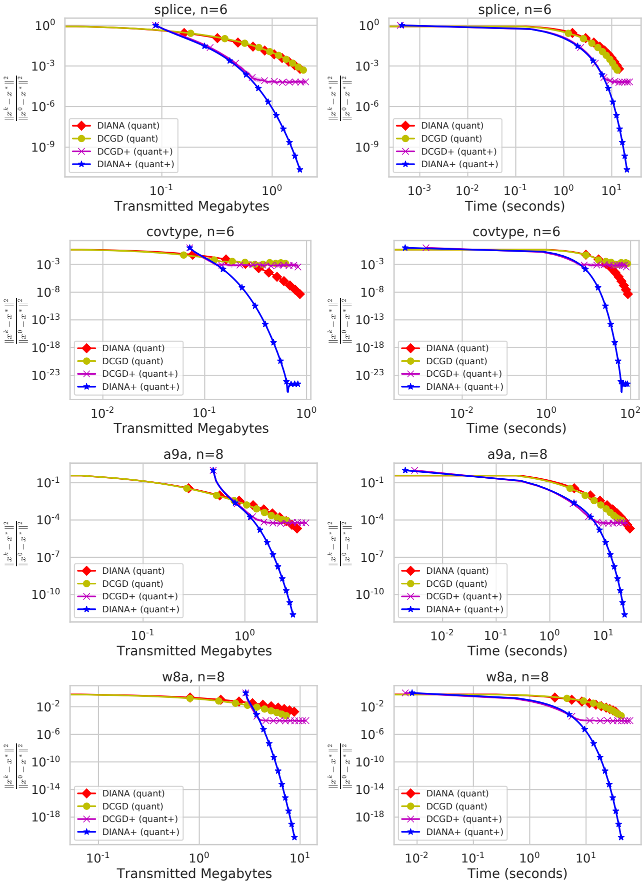

Figure 2. Comparison of smoothness-aware DCGD+/DIANA+ methods with varying-step quantization (`quant+`) to original DCGD/DIANA methods with standard quantization (`quant`). Note that in `quant+` workers need to send $\mathbf{L}_i^{1/2} \in \mathbb{R}^{d\times d}$ and quantization steps $h_i \in \mathbb{R}^d$ to the master before the training. This leads to extra costs in communication bits and time, which are taken into consideration.

6.3. Comparison to matrix-smoothness-aware sparsification

Second experiment is devoted to the performance of three smoothness-aware compression techniques —block quantization (`block quant+`) of Section 4, varying-step quantization (`quant+`) of Section 5 and smoothness-aware sparsification strategy (`rand-`τ`+`) of Safaryan et al. (2021). All three compression techniques are shown to outperform the standard compression strategies by at most $\mathcal{O}(n)$ times in theory. For the sparsification, we use the optimal probabilities and the sampling size $\tau = d/n$ as suggested in Section 5.3 of (Safaryan et al., 2021). The empirical results in Figure 3 illustrate that the varying-step quantization technique (`quant+`) is always better than the smoothness-aware sparsification (Safaryan et al., 2021), in terms of both communication cost and wall-clock time. Our block quantization technique also beats sparsification when the dimension of the model is relatively high.

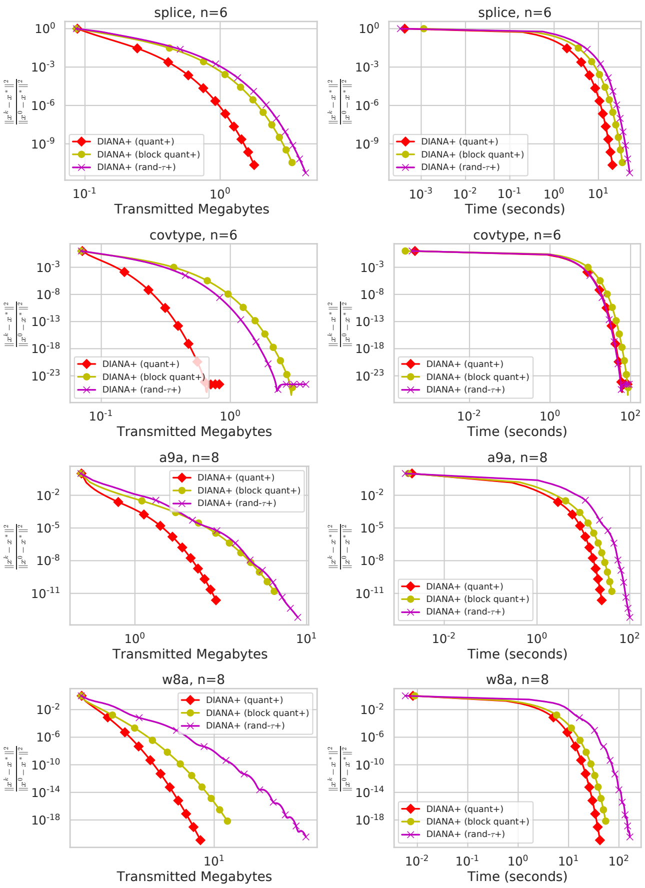

Figure 3. Comparison of three matrix-smoothness-aware compression techniques employed in DIANA+ method: varying-step quantization `quant+`, our variant of block quantization `block quant+`, and smoothness-aware sparsification `rand-`τ`+` of Safaryan et al. (2021).

References

Albasyoni, A., Safaryan, M., Condat, L., and Richtárik, P. Optimal gradient compression for distributed and federated learning. *arXiv preprint arXiv:2010.03246*, 2020.

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pp. 1709–1720, 2017.

Alistarh, D., Hoefler, T., Johansson, M., Konstantinov, N., Khirirat, S., and Renggli, C. The convergence of sparsified gradient methods. In *Advances in Neural Information Processing Systems*, pp. 5977–5987, 2018.

Bekkerman, R., Bilenko, M., and Langford, J. *Scaling up machine learning: Parallel and distributed approaches*. Cambridge University Press, 2011.

Bernstein, J., Wang, Y.-X., Azizzadenesheli, K., and Anandkumar, A. SignSGD: Compressed optimisation for non-convex problems. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 560–569, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Brown et al., T. B. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Chang, C.-C. and Lin, C.-J. LibSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

Dalcín, L., Paz, R., and Storti, M. MPI for Python. *Journal of Parallel and Distributed Computing*, 65(9):1108–1115, 2005.

Faghri, F., Tabrizian, I., Markov, I., Alistarh, D., Roy, D., and Ramezani-Kebrya, A. Adaptive gradient quantization for data-parallel sgd. In *Advances in Neural Information Processing Systems*, 2020.

Goodall, W. M. Television by pulse code modulation. *The Bell System Technical Journal*, 30(1):33–49, Jan 1951. ISSN 0005-8580. doi: 10.1002/j.1538-7305.1951.tb01365.x.

Gorbunov, E., Hanzely, F., and Richtárik, P. A unified theory of SGD: Variance reduction, sampling, quantization and coordinate descent. In *The 23rd International Conference on Artificial Intelligence and Statistics*, 2020.

Gorbunov, E., Burlachenko, K., Li, Z., and Richtárik, P. MARINA: faster non-convex distributed learning with compression. *arXiv preprint arXiv:2102.07845*, 2021.

Hanzely, F. and Richtárik, P. One method to rule them all: Variance reduction for data, parameters and many new methods. preprint arXiv:1905.11266, 2019.

Horváth, S., Kovalev, D., Mishchenko, K., Stich, S., and Richtárik, P. Stochastic distributed learning with gradient quantization and variance reduction. *arXiv preprint arXiv:1904.05115*, 2019.

Khirirat, S., Feyzmahdavian, H. R., and Johansson, M. Distributed learning with compressed gradients. In *arXiv preprint arXiv:1806.06573*, 2018.

Konečný, J. and Richtárik, P. Randomized distributed mean estimation: accuracy vs communication. *Frontiers in Applied Mathematics and Statistics*, 4(62):1–11, 2018.

Li, Z., Kovalev, D., Qian, X., and Richtárik, P. Acceleration for compressed gradient descent in distributed and federated optimization. In *International Conference on Machine Learning*, 2020.

Lin, Y., Han, S., Mao, H., Wang, Y., and Dally, W. J. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *International Conference on Learning Representations*, 2018.

Mishchenko, K., Gorbunov, E., Takáč, M., and Richtárik, P. Distributed learning with compressed gradient differences. In *arXiv preprint arXiv:1901.09269*, 2019.

Mishchenko, K., Wang, B., Kovalev, D., and Richtárik, P. IntSGD: Floatless compression of stochastic gradients. *arXiv preprint arXiv:2102.08374*, 2021.

Narayanan, D., Shoeybi, M., Casper, J., LeGresley, P., Patwary, M., Korthikanti, V., Vainbrand, D., and Catanzaro, B. Scaling language model training to a trillion parameters using megatron. https://developer.nvidia.com/blog/scaling-language-model-training-to-a-trillion-parameters-using-megatron/, 2021.

Nesterov, Y. *Introductory lectures on convex optimization: a basic course.* Kluwer Academic Publishers, 2004.

Ramezani-Kebrya, A., Faghri, F., Markov, I., Aksenov, V., Alistarh, D., and Roy, D. M. NUQSGD: Provably communication-efficient data-parallel SGD via nonuniform quantization. *arXiv preprint arXiv:2104.13818*, 2021.

Roberts, L. Picture coding using pseudo-random noise. *IRE Transactions on Information Theory*, 8(2):145–154, February 1962. ISSN 0096-1000. doi: 10.1109/TIT.1962.1057702.

Safaryan, M. and Richtárik, P. Stochastic sign descent methods: New algorithms and better theory. In *International Conference on Machine Learning (ICML)*, 2021.

Safaryan, M., Hanzely, F., and Richtárik, P. Smoothness matrices beat smoothness constants: Better communication compression techniques for distributed optimization. *arXiv preprint arXiv:2102.07245*, 2021.

Schmidhuber, J. Deep learning in neural networks: An overview. In *Neural networks*, volume 61, pp. 85–117, 2015.

Vaswani, S., Bach, F., and Schmidt, M. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. In *22nd International Conference on Artificial Intelligence and Statistics*, volume 89 of *PMLR*, pp. 1195–1204, 2019.

Vogels, T., Karimireddy, S. P., and Jaggi, M. PowerSGD: Practical low-rank gradient compression for distributed optimization. *arXiv prepring arXiv:1905.13727*, 2019.

Wangni, J., Wang, J., Liu, J., and Zhang, T. Gradient sparsification for communication-efficient distributed optimization. In *Advances in Neural Information Processing Systems*, pp. 1306–1316, 2018.

Zhang, H., Li, J., Kara, K., Alistarh, D., Liu, J., and Zhang, C. ZipML: Training linear models with end-to-end low precision, and a little bit of deep learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 4035–4043, 2017.

Appendix

Contents

A. Conclusions and Limitations

In this work we extended the matrix-smoothness-aware sparsification strategy of Safaryan et al. (2021) to arbitrary unbiased compression schemes. This significantly broadens the use of smoothness matrices in communication efficient distributed methods.

A.1. Generalization and quantization

It is worth to mention that our results generalize those of Safaryan et al. (2021) in a tight manner. That is, we recover the same convergence guarantees as a special case. Indeed, if compression operators $\mathcal{C}_i$ are diagonal sketches $\mathbf{C}_i$ generated independently from others and via arbitrary samplings, then

$$\begin{aligned}
\mathcal{L}_i &= \mathcal{L}(\mathbf{C}_i, \mathbf{L}_i) \\
&= \inf\left\{\mathcal{L} \geq 0 \colon \mathbb{E}\left[\|\mathbf{C}_i x - x\|^2_{\mathbf{L}_i}\right] \leq \mathcal{L}\|x\|^2 \ \forall x \in \mathbb{R}^d\right\} \\
&= \inf\left\{\mathcal{L} \geq 0 \colon x^\top \mathbb{E}\left[(\mathbf{C}_i - \mathbf{I})\mathbf{L}_i(\mathbf{C}_i - \mathbf{I})\right] x \leq \mathcal{L}\|x\|^2 \ \forall x \in \mathbb{R}^d\right\} \\
&= \lambda_{\max}\left(\mathbb{E}\left[(\mathbf{C}_i - \mathbf{I})\mathbf{L}_i(\mathbf{C}_i - \mathbf{I})\right]\right) \\
&= \lambda_{\max}\left(\mathbb{E}\left[\mathbf{C}_i\mathbf{L}_i\mathbf{C}_i\right] - \mathbf{L}_i\right) \\
&= \lambda_{\max}(\overline{\mathbf{P}}_i \circ \mathbf{L}_i - \mathbf{L}_i) \\
&= \lambda_{\max}(\widetilde{\mathbf{P}}_i \circ \mathbf{L}_i),
\end{aligned}$$

with the same probability matrices $\overline{\mathbf{P}}_i$ and $\widetilde{\mathbf{P}}_i$ defined in (Safaryan et al., 2021).

Further, we designed two novel quantization schemes (see Definitions 3 and 4) capable of properly utilizing matrix smoothness information of local loss functions in distributed optimization. We showed that the proposed quantization schemes can significantly outperform the key baselines both in theory and practice.

A.2. Limitations

Next, we discuss main limitations of our work.

- Note while in this paper we redesigned only two methods, DCGD+ and DIANA+, the modifications we suggest are not limited to these two methods and can be applied to other distributed methods. In particular, with a similar proof technique, ADIANA+ method of (Safaryan et al., 2021) introduced with sparsification can also be extended to arbitrary unbiased compression operator using the new notion of $\mathcal{L}(\mathcal{C}, \mathbf{L})$.
- The server is required to store $d \times d$ matrices $\mathbf{L}_i^{1/2}$ for all nodes $i \in [n]$ and multiply them by sparse updates $\mathcal{C}_i^k(\mathbf{L}_i^{\dagger 1/2} \nabla f_i(x^k))$ in each iteration. Moreover, each node i is required to store only its smoothness matrix $\mathbf{L}_i^{\dagger 1/2}$ and perform multiplication $\mathbf{L}_i^{\dagger 1/2} \nabla f_i(x^k)$ in each iterate. Hence, our methods are practical when either dimension d is not too big or smoothness matrices $\mathbf{L}_i$ are of special structure (e.g., diagonal, low-rank).
- For the sake of presentation, we analyzed both DCGD+ and DIANA+ when exact local gradients, ∇f_i, can be computed by all nodes in each iteration. However, we believe that it is possible to extend the analysis to stochastic local gradient oracles. Current tools handling stochastic gradients can be easily applied to our matrix-smoothness-aware compression techniques.
- In our distributed methods we only compress uplink communication from nodes to the server, which is typically more bandwidth limited than downlink communication from the server to nodes. We believe that techniques that ensure compressed communication in both directions can be applied in our setting, too.
- We developed all our theory for strongly convex objectives. Extending the theory to convex and non-convex problems in a tight manner seems to be more challenging.

B. Additional Experiments

In this section we provide additional experiments to highlight effectiveness of our approach.

B.1. Comparison to standard quantization techniques

First, we compare DCGD+/DIANA+ with the block quantization technique (`block quant+`) described in Section 4 to DCGD (Khirirat et al., 2018)/DIANA (Mishchenko et al., 2019) with the standard quantization technique (`quant`) in (Alistarh et al., 2017). As shown in Figure 6, DCGD+ (`block quant+`) and DIANA+ (`block quant+`) outperform DCGD (`quant`) and DIANA (`quant`) when d is larger. This is understandable because the extra cost on communication B norms becomes neglectable when the dimension is relatively high given the number of blocks, where splitting the whole parameters into blocks makes more sense.

Next, we compare DCGD+/DIANA+ with our second quantization technique (`quant+`) that has varying number of quantization steps per coordinate to DCGD (`quant`) and DIANA (`quant`). Figure 7 demonstrates that DCGD+ (`quant+`) and DIANA+ (`quant+`) lead to significant improvement.

B.2. Ablation study of DIANA+ (`block quant+`) and DIANA+ (`quant+`)

As mentioned by (Alistarh et al., 2017), combining DCGD and block quantization can improve its iteration complexity at the cost of transmitting extra $32B$ bits per iteration, which might also lead to better total communication complexity. Thus, the advantage of DIANA+ (`block quant+`) over DIANA (`quant`) may come from either splitting the features into blocks or exploiting the smoothness matrix. To further demistefy the improvement of DIANA+ (block quant+) , we compare the results of DIANA+ (`block quant+`), DIANA+ (`block quant`), DIANA (`block quant`) and DIANA (`quant`) in Figure 5. The difference between `block quant` and `block-quant+` is that the former one uses the same number of quantization levels for different blocks while the latter one uses varying numbers. It can be seen from Figure 5 that DIANA+ (`block-quant+`) consistently outperforms other methods because it optimally exploits the block structure and the smoothness matrix.

We also demonstrate that how DIANA+ perform with varying or fixed number of levels. As seen in Figure 6, the varying number of levels are beneficial on most of the datasets.

B.3. Comparison to matrix-smoothness-aware sparsification

Moreover, we also compare the performance of three smoothness-aware compression techniques —block quantization (`block quant+`) of Section 4, varying-step quantization (`quant+`) of Section 5 and smoothness-aware sparsification strategy (rand-τ+) of Safaryan et al. (2021). All three compression techniques are shown to outperform the standard compression strategies by at most $\mathcal{O}(n)$ times in theory. For the sparsification, we use the optimal probabilities and the sampling size $\tau = d/n$ as suggested in Section 5.3 of (Safaryan et al., 2021). The empirical results in Figure 8 illustrate that the varying-step quantization technique (`quant+`) is always better than the smoothness-aware sparsification (Safaryan et al., 2021), in terms of both communication cost and wall-clock time. Our block quantization technique also beats sparsification when the dimension of the model is relatively high.

Table 2. Information of the experiments on ℓ_2-regularized logistic regression.

Dataset	#Instances N	Dimension d	#Workers n	#Instances/worker m
german	1,000	24	4	250
svmguide3	1,243	21	4	310
covtype	581,012	54	6	145,253
splice	1,000	60	6	166
w8a	49,749	300	8	6,218
a9a	22,696	123	8	2,837

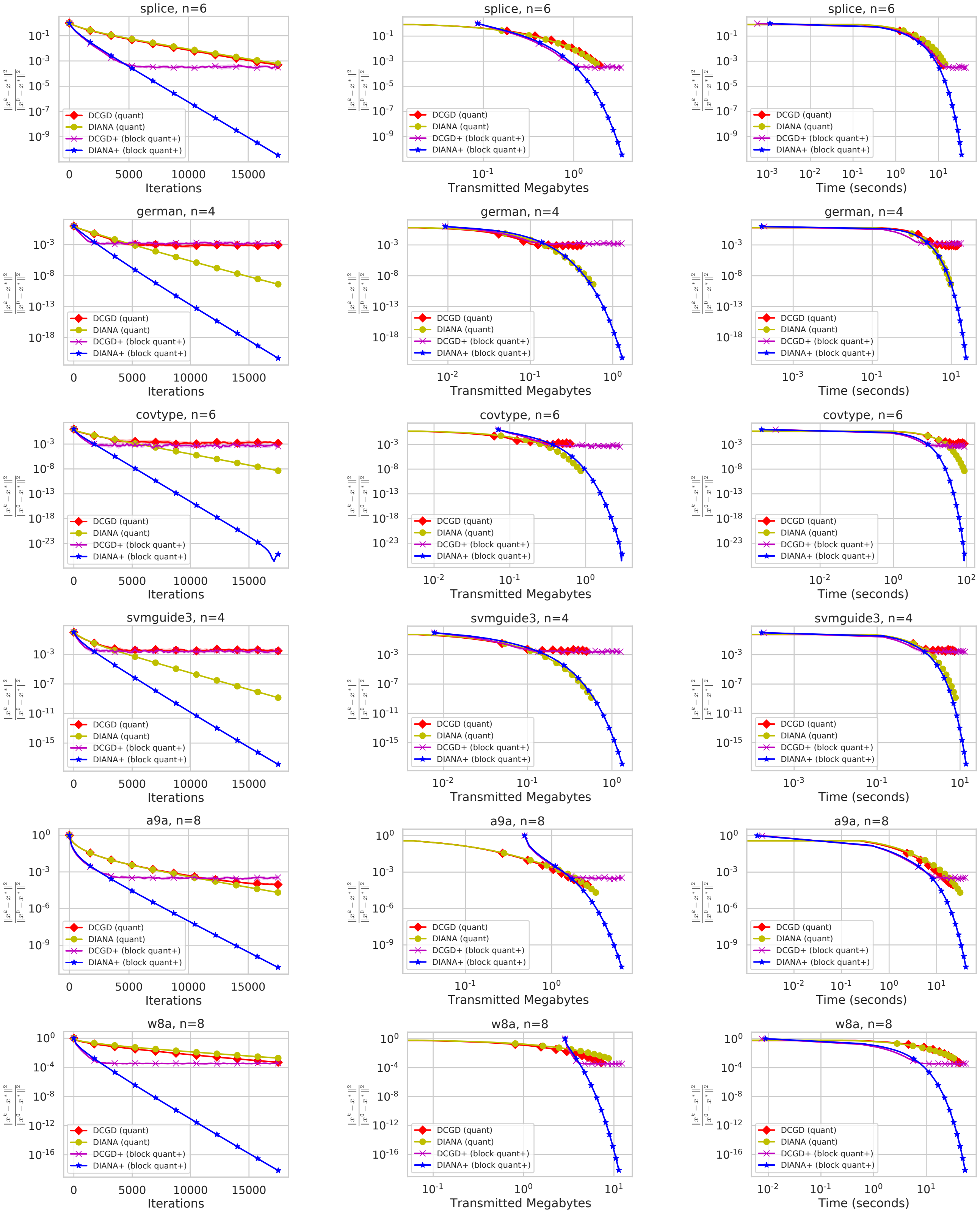

Figure 4. Comparison of DCGD+ (`block quant+`) and DIANA+ (`block quant+`) with DCGD (`quant`) and DIANA (`quant`).

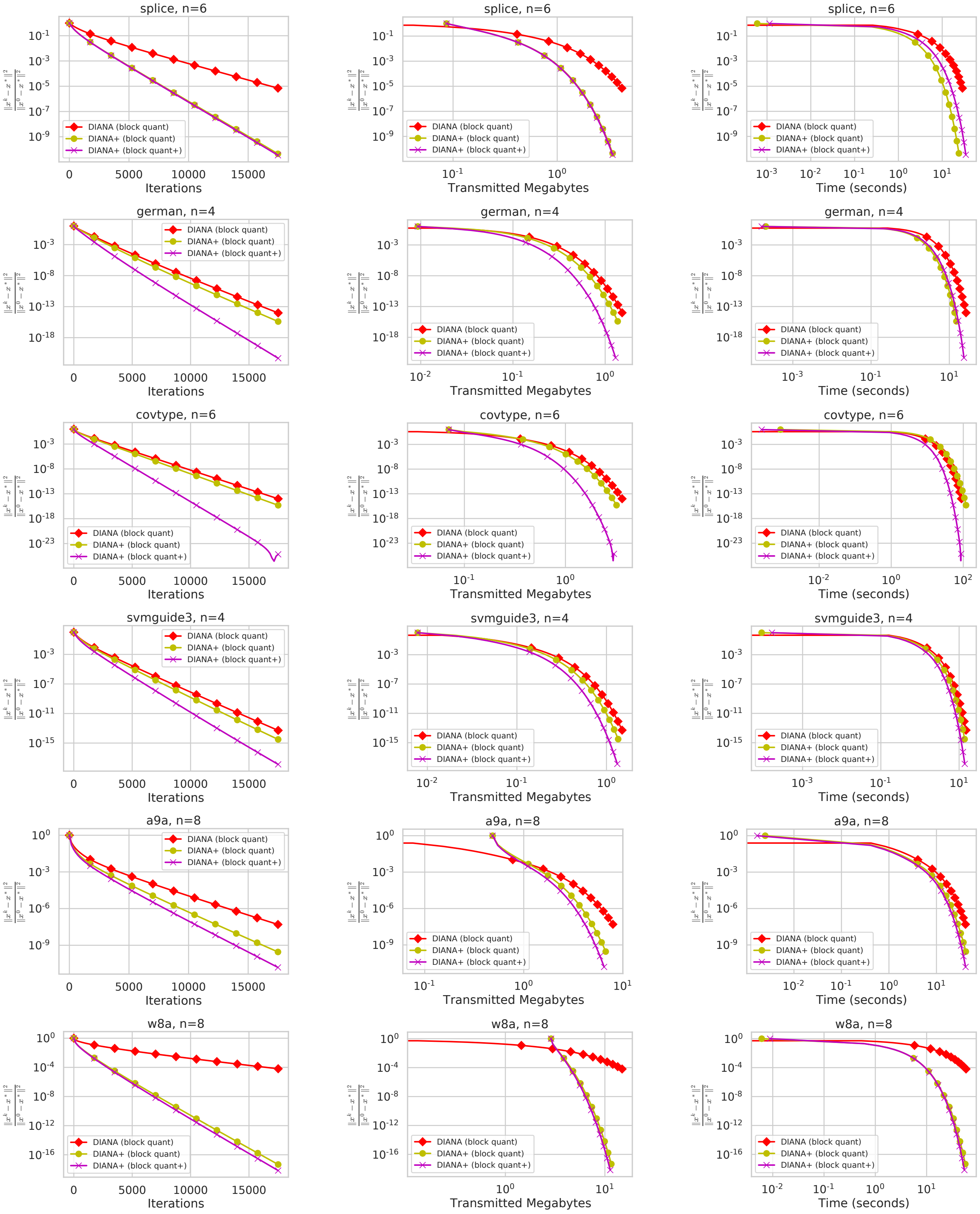

Figure 5. Comparison of DIANA+ (`block quant+`), DIANA+ (`block quant`), DIANA (`block quant`) and DIANA (`quant`).

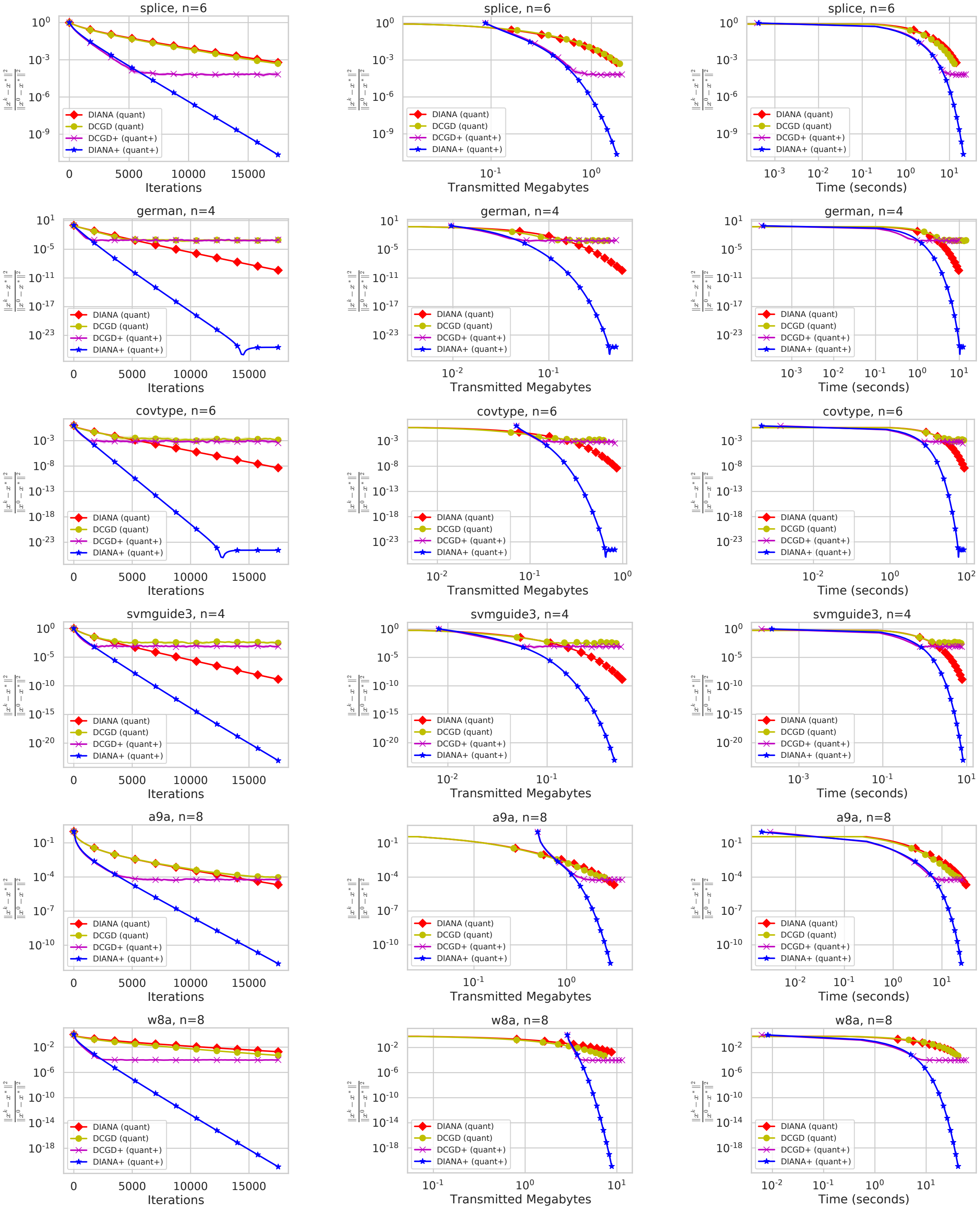

Figure 6. Comparison of DCGD+ (`quant+`) and DIANA+ (`quant+`) with DCGD (`quant`) and DIANA (`quant`).

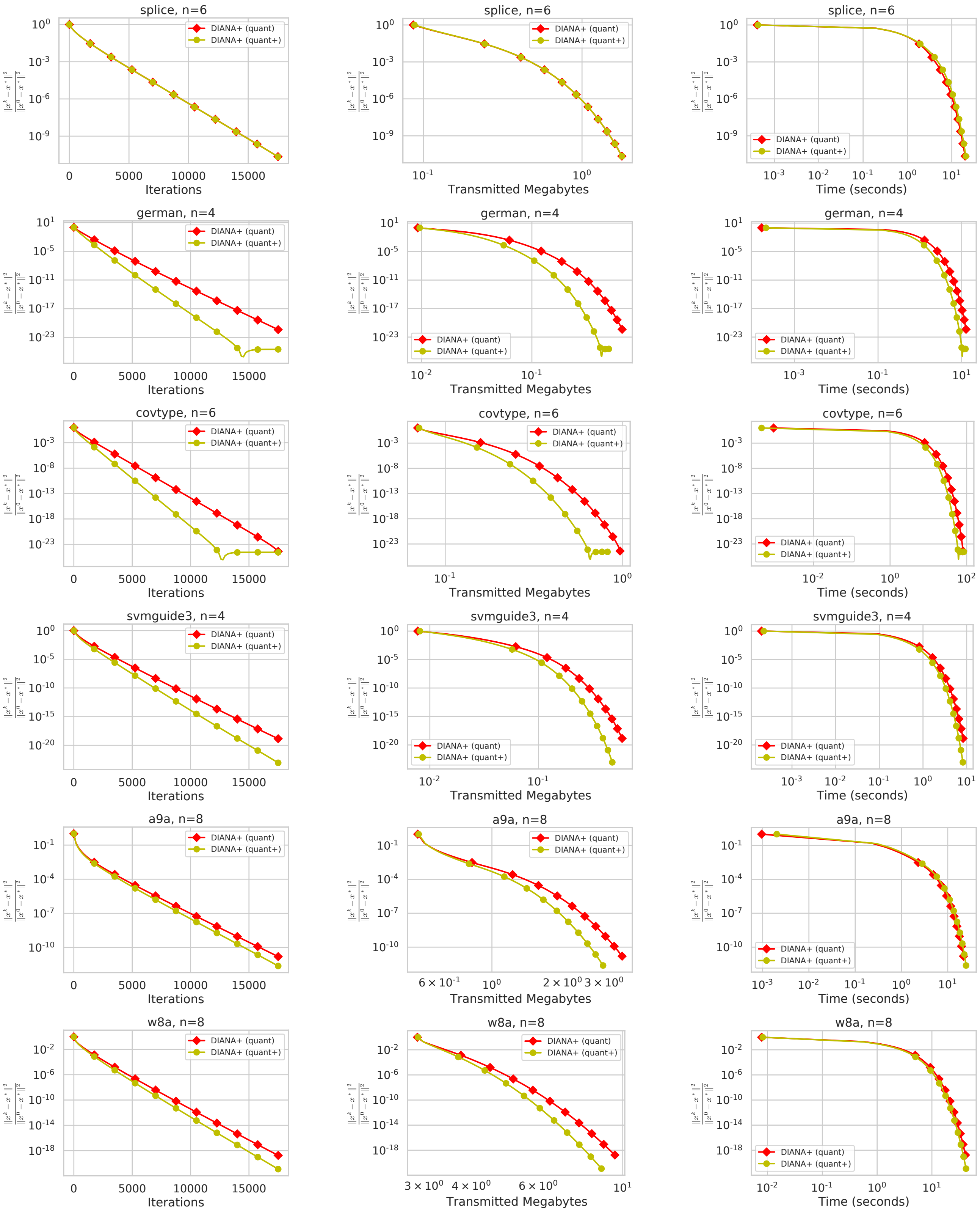

Figure 7. Comparison of DIANA+ with quantization that has varying or fixed number of levels.

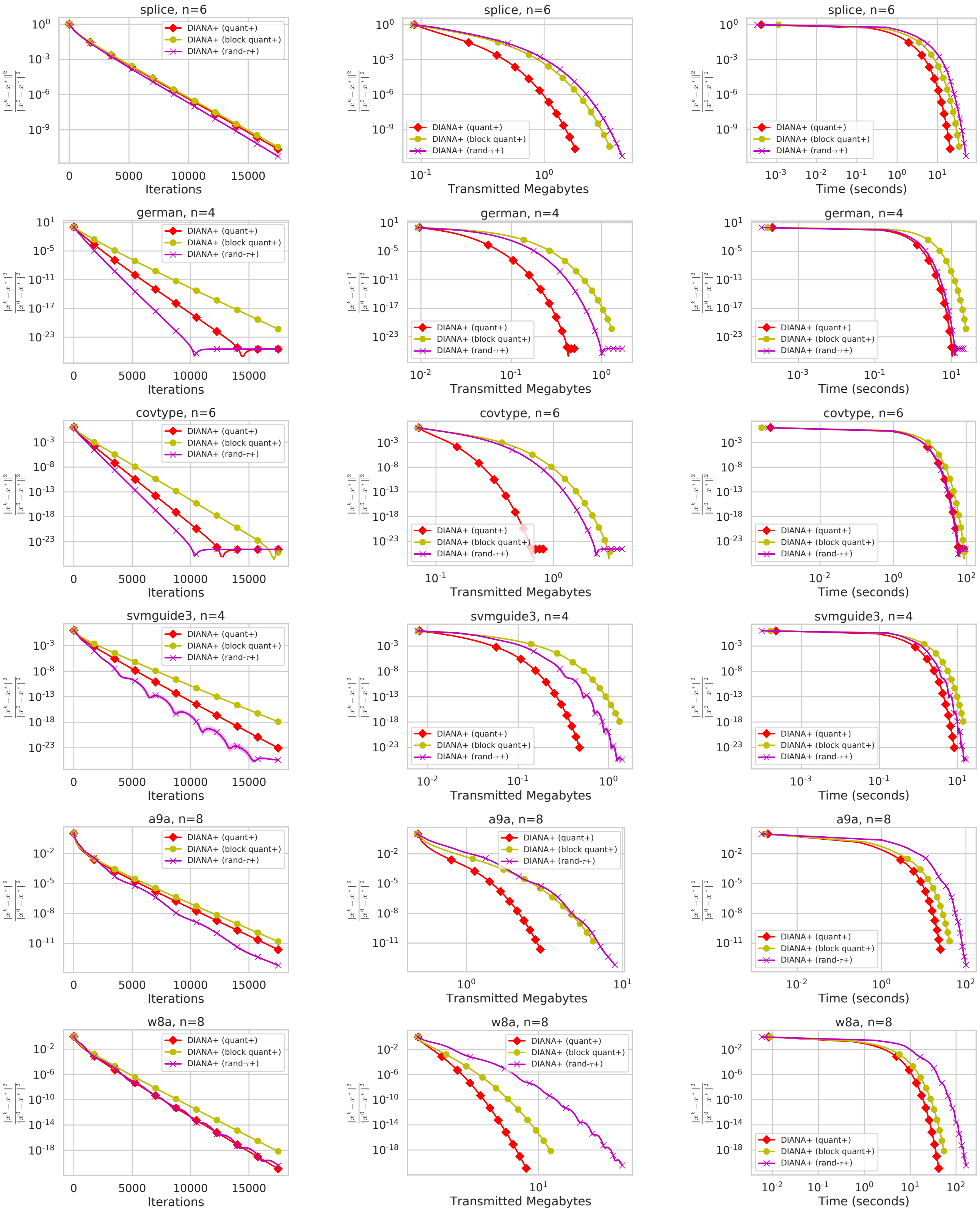

Figure 8. Comparison of smoothness-aware DCGD+/DIANA+ methods with varying-step quantization (quant+) to original DCGD/DIANA methods with standard quantization (quant). Note that in quant+ workers need to send $\mathbf{L}_i^{1/2} \in \mathbb{R}^{d\times d}$ and quantization steps $h_i \in \mathbb{R}^d$ to the master before the training. This leads to extra costs in communication bits and time, which are taken into consideration.

C. Proofs for Section 3: Smoothness-Aware Distributed Methods with General Compressors

Here we provide the proofs of Theorem 1 and Theorem 2. Both proofs follow similar steps done for sparsification in (Safaryan et al., 2021).

C.1. Proof of Theorem 1: DCGD+ with arbitrary unbiased compression

To simplify the notation, let us skip the iteration count k in the derivations. We are going to estimate the quantity $\mathbb{E}\left[\|g(x)-\nabla f(x^*)\|^2\right]$ and establish the following bound for the gradient estimator $g(x)=\frac{1}{n}\sum_{i=1}^n \mathbf{L}_i^{1/2}\mathcal{C}_i(\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x))$:

$$\mathbb{E}\left[\|g(x)-\nabla f(x^*)\|^2\right] \le 2\left(L+\frac{2\mathcal{L}_{\max}}{n}\right)D_f(x,x^*)+\frac{2\sigma_+^*}{n}.$$

Due to Lemma E.3 (Hanzely & Richtárik, 2019), we have $\nabla f_i(x)=\mathbf{L}_i^{1/2}r_i$ for some r_i. Therefore,

$$\mathbb{E}\left[\mathbf{L}_i^{1/2}\mathcal{C}_i(\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2}r_i)\right]=\mathbf{L}_i^{1/2}\mathbb{E}\left[\mathcal{C}_i(\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2}r_i)\right]=\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2}r_i=\mathbf{L}_i^{1/2}r_i=\nabla f_i(x), \tag{20}$$

which implies unbiasedness of the estimator $g(x)$, namely $\mathbb{E}\left[g(x)\right]=\nabla f(x)$. Next, note that

$$\begin{aligned}
\mathbb{E}\left[\|g(x)-\nabla f(x)\|^2\right] &= \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \mathbf{L}_i^{1/2}\mathcal{C}_i(\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x))-\nabla f_i(x)\right\|^2\right]\\
&= \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left[\left\|\mathbf{L}_i^{1/2}\mathcal{C}_i(\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x))-\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x)\right\|^2\right]\\
&= \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left[\left\|\mathcal{C}_i(\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x))-\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x)\right\|^2_{\mathbf{L}_i}\right]\\
&\le \frac{1}{n^2}\sum_{i=1}^n \mathcal{L}(\mathcal{C}_i,\mathbf{L}_i)\|\nabla f_i(x)\|^2_{\mathbf{L}_i^\dagger}\\
&\le \frac{2}{n^2}\sum_{i=1}^n \mathcal{L}_i\|\nabla f_i(x)-\nabla f_i(x^*)\|^2_{\mathbf{L}_i^\dagger}+\frac{2}{n^2}\sum_{i=1}^n \mathcal{L}_i\|\nabla f_i(x^*)\|^2_{\mathbf{L}_i^\dagger}\\
&\le \frac{4}{n^2}\sum_{i=1}^n \mathcal{L}_i D_{f_i}(x,x^*)+\frac{2\sigma_+^*}{n}\\
&\le \frac{4\mathcal{L}_{\max}}{n}D_f(x,x^*)+\frac{2\sigma_+^*}{n},
\end{aligned}$$

which together with convexity and L-smoothness of f implies

$$\begin{aligned}
\mathbb{E}\left[\|g(x)-\nabla f(x^*)\|^2\right] &= \|\nabla f(x)-\nabla f(x^*)\|^2+\mathbb{E}\left[\|g(x)-\nabla f(x)\|^2\right]\\
&\le 2LD_f(x,x^*)+\frac{4\mathcal{L}_{\max}}{n}D_f(x,x^*)+\frac{2\sigma_+^*}{n}\\
&\le 2\left(L+\frac{2\mathcal{L}_{\max}}{n}\right)D_f(x,x^*)+\frac{2\sigma_+^*}{n}.
\end{aligned}$$

Applying the result of Gorbunov et al. (2020) we conclude the proof.

C.2. Proof of Theorem 2: DIANA+ with arbitrary unbiased compression

We start with the unbiasedness of the estimator

$$g^k=\frac{1}{n}\sum_{i=1}^n \mathbf{L}_i^{1/2}\mathcal{C}_i\left(\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x)-u_i^k)\right)+u_i^k.$$

In (20), we showed unbiasedness using inclusion $\nabla f_i(x^k) \in \mathrm{range}(\mathbf{L}_i)$. Assuming $u_i^k \in \mathrm{range}(\mathbf{L}_i)$ for all $k \geq 0$, we get $\nabla f_i(x^k) - u_i^k \in \mathrm{range}(\mathbf{L}_i)$ for all $k \geq 0$. Hence, in the same way we can show unbiasedness of g^k as

$$\begin{aligned}
\mathbb{E}_k\left[g^k\right] &= \frac{1}{n}\sum_{i=1}^n \mathbf{L}_i^{1/2}\mathbb{E}_k\left[\mathcal{C}_i\left(\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x) - u_i^k)\right)\right] + u_i^k \\
&= \frac{1}{n}\sum_{i=1}^n \mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x) - u_i^k) + u_i^k \\
&= \frac{1}{n}\sum_{i=1}^n \nabla f_i(x^k) = \nabla f(x^k).
\end{aligned}$$

The inclusion $u_i^k \in \mathrm{range}(\mathbf{L}_i)$ directly follows from the initialization $u_i^0 \in \mathrm{range}(\mathbf{L}_i)$ (see line 1 of Algorithm 2) and linear update rule of $u_i^{k+1} = u_i^k + \alpha \mathbf{L}_i^{1/2}\Delta_i^k$ (see line 5 of Algorithm 2). As both $\nabla f_i(x^k)$ and u_i^k belong to $\mathrm{range}(\mathbf{L}_i)$, denote $\nabla f_i(x^k) - u_i^k = \mathbf{L}_i^{1/2} r_i^k$. Next we bound

$$\begin{aligned}
\mathbb{E}\left[\|g(x) - \nabla f(x)\|^2\right] &= \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \mathbf{L}_i^{1/2}\mathcal{C}_i\left(\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x) - u_i^k)\right) + u_i^k - \nabla f_i(x)\right\|^2\right] \\
&= \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left[\left\|\mathbf{L}_i^{1/2}\mathcal{C}_i^k\left(\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x) - u_i^k)\right) - \mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x) - u_i^k)\right\|^2\right] \\
&= \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left[\left\|\mathcal{C}_i^k\left(\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x) - u_i^k)\right) - \mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x) - u_i^k)\right\|_{\mathbf{L}_i}^2\right] \\
&\leq \frac{1}{n^2}\sum_{i=1}^n \mathcal{L}(\mathcal{C}_i, \mathbf{L}_i)\|\nabla f_i(x) - u_i^k\|_{\mathbf{L}_i^\dagger}^2 \\
&\leq \frac{2\mathcal{L}_{\max}}{n^2}\sum_{i=1}^n \|\nabla f_i(x) - \nabla f_i(x^*)\|_{\mathbf{L}_i^\dagger}^2 + \frac{2\mathcal{L}_{\max}}{n^2}\sum_{i=1}^n \|u_i^k - \nabla f_i(x^*)\|_{\mathbf{L}_i^\dagger}^2 \\
&\leq \frac{4\mathcal{L}_{\max}}{n^2}\sum_{i=1}^n D_{f_i}(x, x^*) + \frac{2\mathcal{L}_{\max}}{n}\sigma_+^k \\
&= \frac{4\mathcal{L}_{\max}}{n} D_f(x, x^*) + \frac{2\mathcal{L}_{\max}}{n}\sigma_+^k,
\end{aligned}$$

where $\sigma_+^k \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n \|u_i^k - \nabla f_i(x^*)\|_{\mathbf{L}_i^\dagger}^2$ is the error in the gradient learning process. To proceed, we need to establish contractive recurrence relation for σ_+^k. For each summand, we have

$$
\begin{aligned}
& \mathbb{E}_k\left[\left\|u_i^{k+1} - \nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger}\right] \\
= \;& \mathbb{E}_k\left[\left\|u_i^{k} - \nabla f_i(x^*) + \alpha \overline{\Delta}_i^k\right\|^2_{\mathbf{L}_i^\dagger}\right] \\
= \;& \left\|u_i^{k} - \nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger} + 2\alpha \left\langle u_i^k - \nabla f_i(x^*), \nabla f_i(x^k) - u_i^k \right\rangle_{\mathbf{L}_i^\dagger} + \alpha^2 \mathbb{E}\left[\left\|\mathbf{L}_i^{1/2}\mathcal{C}_i\left(\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x) - u_i^k)\right)\right\|^2_{\mathbf{L}_i^\dagger}\right] \\
\leq \;& \left\|u_i^{k} - \nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger} + 2\alpha \left\langle u_i^k - \nabla f_i(x^*), \nabla f_i(x^k) - u_i^k \right\rangle_{\mathbf{L}_i^\dagger} + \alpha^2 \mathbb{E}\left[\left\|\mathcal{C}_i\left(\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x) - u_i^k)\right)\right\|^2\right] \\
\leq \;& \left\|u_i^{k} - \nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger} + 2\alpha \left\langle u_i^k - \nabla f_i(x^*), \nabla f_i(x^k) - u_i^k \right\rangle_{\mathbf{L}_i^\dagger} + \alpha^2(1+\omega_i)\left\|\nabla f_i(x^k) - u_i^k\right\|^2_{\mathbf{L}_i^\dagger} \\
\leq \;& \left\|u_i^{k} - \nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger} + 2\alpha \left\langle u_i^k - \nabla f_i(x^*), \nabla f_i(x^k) - u_i^k \right\rangle_{\mathbf{L}_i^\dagger} + \alpha\left\|\nabla f_i(x^k) - u_i^k\right\|^2_{\mathbf{L}_i^\dagger} \\
= \;& (1-\alpha)\left\|u_i^{k} - \nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger} + \alpha\left\|\nabla f_i(x^k) - \nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger}, \\
\leq \;& (1-\alpha)\left\|u_i^{k} - \nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger} + 2\alpha D_{f_i}(x^k, x^*),
\end{aligned}
$$

where we used bounds $\alpha \leq \frac{1}{1+\omega_i}$ and $0 \preceq \mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2} \preceq \mathbf{I}$. Thus, with $\alpha \leq \frac{1}{1+\omega_{\max}}$, the estimator g^k of DIANA+ satisfies

$$
\begin{aligned}
\mathbb{E}_k\left[g^k\right] &= \nabla f(x^k) \\
\mathbb{E}_k\left[\|g^k - \nabla f(x^*)\|^2\right] &\leq 2\left(L + \frac{2\mathcal{L}_{\max}}{n}\right) D_f(x^k, x^*) + \frac{2\mathcal{L}_{\max}}{n}\sigma_+^k \\
\mathbb{E}_k\left[\sigma_+^{k+1}\right] &\leq (1-\alpha)\sigma_+^k + 2\alpha D_f(x^k, x^*).
\end{aligned}
$$

Again, we apply the generic result of Gorbunov et al. (2020) to complete the proof.

D. Proofs for Section 4: Block Quantization

Here we provide the missing proofs of Section 4.

D.1. Proof of the variance bound (10)

Using Definition 3 of compression operator $\mathcal{Q}_h^B$, we have

$$\begin{aligned}
\mathbb{E}\left[\|\mathcal{Q}_h^B(x)-x\|_{\mathbf{L}}^2\right] &= \sum_{l=1}^{B}\|x^l\|^2\mathbb{E}\left[\left\|\xi_l\left(\frac{|x^l|}{\|x^l\|}\right)-\frac{|x^l|}{\|x^l\|}\right\|_{\mathbf{L}^{ll}}^2\right] \\
&\le \sum_{l=1}^{B}\|x^l\|^2\min\left(h_l^2\sum_{j=1}^{d_l}\mathbf{L}_{jj}^{ll}, h_l\sqrt{\sum_{j=1}^{d_l}\left[\mathbf{L}_{jj}^{ll}\right]^2}\right) \\
&\le \max_{1\le l\le B}\min\left(h_l^2\sum_{j=1}^{d_l}\mathbf{L}_{jj}^{ll}, h_l\sqrt{\sum_{j=1}^{d_l}\left[\mathbf{L}_{jj}^{ll}\right]^2}\right)\|x\|^2 \\
&= \max_{1\le l\le B}\min\left(h_l^2\|\,\mathbf{Diag}(\mathbf{L}^{ll})\|_1, h_l\|\,\mathbf{Diag}(\mathbf{L}^{ll})\|\right)\|x\|^2.
\end{aligned}$$

From the definition of $\mathcal{L}(\mathcal{Q}_h^B,\mathbf{L})$ we get

$$\mathcal{L}(\mathcal{Q}_h^B,\mathbf{L}) \le \max_{1\le l\le B}\min\left(h_l^2\|\,\mathbf{Diag}(\mathbf{L}^{ll})\|_1, h_l\|\,\mathbf{Diag}(\mathbf{L}^{ll})\|\right),$$

which implies (10) if we ignore the first term.

D.2. Proof of Theorem 3: DCGD+ with block quantization

First, recall that quantization steps h_i are given by

$$h_{i,l} = \frac{\delta_{i,B}}{\|\,\mathbf{Diag}(\mathbf{L}_i^{ll})\|},\ l\in[B], \quad \text{where } \delta_{i,B} \le \frac{d}{\beta-B}T_{i,B} + \sqrt{\frac{d}{\beta-B}}T_{i,1}.$$

Then, we have

$$\begin{aligned}
\frac{\mathcal{L}_{\max}}{n} &= \frac{1}{n}\max_{i\in[n]}\mathcal{L}(\mathcal{Q}_{h_i}^B,\mathbf{L}_i) \\
&\le \frac{1}{n}\max_{i\in[n]}\delta_{i,B} \\
&\le \frac{1}{n}\max_{i\in[n]}\left[\frac{d}{\beta-B}T_{i,B} + \sqrt{\frac{d}{\beta-B}}T_{i,1}\right] \\
&\le \left[\frac{d/n}{\beta-B}\right]\max_{i\in[n]}T_{i,B} + \sqrt{\frac{d/n}{\beta-B}}\max_{i\in[n]}\frac{T_{i,1}}{\sqrt{n}}.
\end{aligned}$$

Set $\beta = {}^{d}\!/\!{}_{n} + n$ and $B = n$. Since $n = \mathcal{O}(\sqrt{d})$, we have $\beta = \mathcal{O}({}^{d}\!/\!{}_{n})$ and hence $\frac{d/n}{\beta - B} = 1$. For the sake of simplicity, assume $d_l = {}^{d}\!/\!{}_{n}$. Next

$$\begin{aligned}
\frac{T_{i,1}}{\sqrt{n}} &\le \frac{1}{\sqrt{nd}} \sum_{j=1}^{d} \mathbf{L}_{i;jj} \le \frac{\nu_1 L_{\max}}{\sqrt{nd}} \\
T_{i,n} &\le \frac{1}{d} \sum_{l=1}^{n} \sqrt{d_l} \sum_{j=1}^{d_l} \mathbf{L}_{jj}^{ll} \\
&= \frac{\max_{l\in[n]} \sqrt{d_l}}{d} \sum_{j=1}^{d} \mathbf{L}_{jj} \\
&= \frac{\max_{l\in[n]} \sqrt{d_l}}{d} \nu_1 L_{\max} \le \frac{\nu_1 L_{\max}}{\sqrt{nd}}.
\end{aligned}$$

Regardless of the choice h_i, using the following inequalities with respect to matrix order

$$\mathbf{L} \preceq \frac{1}{n} \sum_{i=1}^{n} \mathbf{L}_i, \quad \mathbf{L}_i \preceq n\mathbf{L}, \tag{21}$$

we bound L as follows

$$L = \lambda_{\max}(\mathbf{L}) \overset{(21)}{\le} \lambda_{\max}\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{L}_i\right) \le \frac{1}{n}\sum_{i=1}^{n} \lambda_{\max}(\mathbf{L}_i) = \frac{1}{n}\sum_{i=1}^{n} L_i \overset{(8)}{\le} \frac{\nu}{n} L_{\max}. \tag{22}$$

Hence

$$\frac{L}{\mu} + \frac{\mathcal{L}_{\max}}{\mu n} \le \frac{\nu}{n}\frac{L_{\max}}{\mu} + \frac{2\nu_1}{\sqrt{nd}}\frac{L_{\max}}{\mu} = \mathcal{O}\left(\frac{1}{n}\frac{L_{\max}}{\mu}\right),$$

which guarantees n times fewer communication rounds with the same number of bits per round. In other words, each node communicates $\mathcal{O}({}^{d}\!/\!{}_{n})$ bits to the master in each iteration, which gives us $\mathcal{O}(d)$ communication per communication round. Thus, overall communication complexity to achieve $\varepsilon > 0$ accuracy is

$$\mathcal{O}\left(\frac{d}{n}\frac{L_{\max}}{\mu}\log\frac{1}{\varepsilon}\right).$$

D.3. Proof of Theorem 4: DIANA+ with block quantization

As already mentioned, for DIANA+ each node aims to minimize $\omega_i + \frac{1}{n\mu}\mathcal{L}(\mathcal{Q}_{h_i}^B, \mathbf{L}_i)$ with respect to its quantization steps h_i. Notice that

$$\begin{aligned}
\omega_i + \frac{1}{n\mu}\mathcal{L}(\mathcal{Q}_{h_i}^B, \mathbf{L}_i) &\le \max_{l\in[B]} h_{i,l}\sqrt{d_l} + \max_{l\in[B]} \frac{h_{i,l}}{\mu n} \|\mathbf{Diag}(\mathbf{L}_i^{ll})\| \\
&\le 2\max_{l\in[B]} h_{i,l}\left(\sqrt{d_l} + \frac{1}{\mu n}\|\mathbf{Diag}(\mathbf{L}_i^{ll})\|\right).
\end{aligned}$$

This leads to the following optimization problem with respect to h:

$$\begin{aligned}
\min_{h\in\mathbb{R}^B} \quad & \max_{1\le l\le B} h_l\left(\sqrt{d_l} + \frac{1}{\mu n}\|\mathbf{Diag}(\mathbf{L}_i^{ll})\|\right) \\
\text{s.t.} \quad & \sum_{l=1}^{B}\left(\frac{1}{h_l^2} + \frac{\sqrt{d_l}}{h_l}\right) + B = \beta, \ h_l > 0.
\end{aligned} \tag{23}$$

which is solved similar to (11). Denote

$$A_{il} \overset{\text{def}}{=} \sqrt{d_l} + \frac{1}{\mu n}\| \mathbf{Diag}(\mathbf{L}_i^{ll})\|, \quad \widetilde{T}_{iB} \overset{\text{def}}{=} \frac{1}{d}\sum_{l=1}^{B} \sqrt{d_l} A_{il}.$$

Analogous to (11), the solution of (23) has the following form

$$h_{il} = \frac{\widetilde{\delta}_{iB}}{A_{il}},\ l \in [B],$$

where $\widetilde{\delta}_{iB}$ is determined by the constraint equality of (23) as

$$\widetilde{\delta}_{iB} = \frac{d\widetilde{T}_{i,B}}{2(\beta - B)} + \sqrt{\frac{d^2\widetilde{T}^2_{i,B}}{4(\beta-B)^2} + \frac{d\widetilde{T}^2_{i,1}}{\beta - B}} \le \frac{d}{\beta - B}\widetilde{T}_{i,B} + \sqrt{\frac{d}{\beta - B}}\widetilde{T}_{i,1}.$$

Let us estimate $\widetilde{T}_{i,1}$ and $\widetilde{T}_{i,n}$ using the assumptions $B = n$ and (for the sake of simplicity) $d_l = {}^{d}\!/_{n}$.

$$\begin{aligned}
\widetilde{T}_{i1} &= \frac{1}{\sqrt{d}}\left(\sqrt{d} + \frac{1}{\mu n}\| \mathbf{Diag}(\mathbf{L}_i)\|\right) = 1 + \frac{1}{\mu n\sqrt{d}}\sum_{j=1}^{d}\mathbf{L}_{i;jj} \le 1 + \frac{\nu_1 L_{\max}}{\mu n\sqrt{d}} \\
\widetilde{T}_{in} &= \frac{1}{d}\sum_{l=1}^{n}\sqrt{\frac{d}{n}}\left(\sqrt{\frac{d}{n}} + \frac{1}{\mu n}\| \mathbf{Diag}(\mathbf{L}_i^{ll})\|\right) = 1 + \frac{1}{\mu n\sqrt{nd}}\sum_{l=1}^{n}\| \mathbf{Diag}(\mathbf{L}_i^{ll})\| \\
&\le 1 + \frac{1}{\mu n\sqrt{nd}}\sum_{j=1}^{d}\mathbf{L}_{i;jj} = 1 + \frac{\nu_1 L_{\max}}{\mu n\sqrt{nd}}.
\end{aligned}$$

Next, using $\beta = {}^{d}\!/_{n} + n$ and $\nu_1 = \mathcal{O}(1)$, we get

$$\begin{aligned}
\omega_i + \frac{1}{n\mu}\mathcal{L}(\mathcal{Q}^n_{h_i}, \mathbf{L}_i) &\le 2\widetilde{\delta}_{in} \\
&\le \frac{2d}{\beta - n}\widetilde{T}_{in} + 2\sqrt{\frac{d}{\beta - n}}\widetilde{T}_{i1} \\
&= 2n\widetilde{T}_{in} + 2\sqrt{n}\widetilde{T}_{i1} \\
&\le 2n\left(1 + \frac{\nu_1 L_{\max}}{\mu n\sqrt{nd}}\right) + 2\sqrt{n}\left(1 + \frac{\nu_1 L_{\max}}{\mu n\sqrt{d}}\right) = \mathcal{O}\left(n + \frac{1}{\sqrt{nd}}\frac{L_{\max}}{\mu}\right).
\end{aligned}$$

Together with (22), we complete the proof with the following iteration complexity:

$$\mathcal{O}\left(n + \frac{1}{n}\frac{L_{\max}}{\mu} + \frac{1}{\sqrt{nd}}\frac{L_{\max}}{\mu}\right).$$

E. Proofs for Section 5: Quantization with varying steps

In this part of the appendix we provide missing proofs and detailed arguments of Section 5.

E.1. An encoding scheme for $\mathcal{Q}_h$ operator

To communicate a vector of the form $\mathcal{Q}_h(x)$, we adapt the encoding scheme of Albasyoni et al. (2020). From the definition, we have

$$[\mathcal{Q}_h(x)]_j = \|x\| \cdot \mathrm{sign}(x_j \hat{k}_j) \cdot \hat{k}_j h_j$$

for all $j \in [d]$, where $\hat{k}_j \geq 0$ are non-negative random variables coming from (9). Thus, we need to encode the magnitude $\|x\|$, signs $\mathrm{sign}(x_j \hat{k}_j)$ and non-negative integers $\hat{k}_j$.

For the magnitude $\|x\|$ we need just 31 bits. Let $n_0 \overset{\text{def}}{=} |\{j \in [d] \colon \hat{k}_j = 0\}|$ be the number of coordinates x_j that are compressed to 0. To communicate signs $\{\mathrm{sign}(x_j \hat{k}_j) \colon j \in [d]\}$, we first send the locations of those n_0 coordinates and then $d - n_0$ bits for the values ± 1. Sending n_0 positions can be done by sending $\log d$ bits representing the number n_0, followed by $\log \binom{d}{n_0}$ bits for the positions. For the signs, we need $\log d + \log \binom{d}{\hat{n}_0} + d - \hat{n}_0 \leq \log d + d \log 3$ bits at most. Finally, it remains to encode $\hat{k}_j$'s for which we only need to send nonzero entries since the positions of $\hat{k}_j = 0$ are already encoded. We encode $\hat{k}_j \geq 1$ with $\hat{k}_j$ bits: $\hat{k}_j - 1$ ones followed by 0. Hence, the expected number of bits to encode $\hat{k}_j$'s is

$$\mathbb{E}\left[\sum_{j=1}^d \hat{k}_j\right] \overset{(9)}{=} \sum_{j=1}^d \frac{v_j}{h_j} \leq \sqrt{\sum_{j=1}^d v_j^2}\sqrt{\sum_{j=1}^d \frac{1}{h_j^2}} = \sqrt{\sum_{j=1}^d \frac{1}{h_j^2}} = \|h^{-1}\|,$$

where $v_j = \frac{|x_j|}{\|x\|}$.

In total, $\mathcal{Q}_h(x)$ can be encoded by

$$31 + \log d + \log \binom{d}{\hat{n}_0} + d - \hat{n}_0 + \|h^{-1}\|$$

bits. Lastly, the $\log \binom{d}{\hat{n}_0}$ term can be upper bounded by the binary entropy function $H_2(t) \overset{\text{def}}{=} -t \log t - (1-t)\log(1-t)$ (see (Albasyoni et al., 2020) for more details), and the expected number of encoding bits for $\mathcal{Q}_h(x)$ can be upper bounded by

$$31 + \log d + dH_2\left(\frac{\|\hat{x}\|_0}{d}\right) + \|\hat{x}\|_0 + \|h^{-1}\|,$$

where $\hat{x} = \mathcal{Q}_h(x)$.

E.2. Proof of the variance bound (15)

Let $v \in \mathbb{R}^d$ be the unit vector with non-negative entries $v_j = {}^{|x_j|}/_{\|x\|}$ for $j \in [d]$. Then

$$\begin{aligned}
\mathbb{E}\left[\|\mathcal{Q}_h(x) - x\|_{\mathbf{L}}^2\right] &= \mathbb{E}\left[\left\|\|x\| \cdot \operatorname{sign}(x) \cdot \xi\left(\frac{|x|}{\|x\|}\right) - \|x\| \cdot \operatorname{sign}(x) \cdot \frac{|x|}{\|x\|}\right\|_{\mathbf{L}}^2\right] \\
&= \|x\|^2 \mathbb{E}\left[\|\xi(v) - v\|_{\mathbf{L}}^2\right] \\
&= \|x\|^2 \mathbb{E}\left[\sum_{j,l=1}^d \mathbf{L}_{jl}\left(\xi_j(v_j) - v_j\right)\left(\xi_l(v_l) - v_l\right)\right] \\
&= \|x\|^2 \sum_{j=1}^d \mathbf{L}_{jj} \mathbb{E}\left[\left(\xi_j(v_j) - v_j\right)^2\right] \qquad (24) \\
&= \|x\|^2 \sum_{j=1}^d \mathbf{L}_{jj}\left(v_j - k_j h_j\right)\left((k_j+1)h_j - v_j\right) \\
&= \|x\|^2 \sum_{j=1}^d \mathbf{L}_{jj} h_j^2 \left(\frac{v_j}{h_j} - k_j\right)\left[1 - \left(\frac{v_j}{h_j} - k_j\right)\right] \\
&\leq \|x\|^2 \sum_{j=1}^d \mathbf{L}_{jj} h_j^2 \min\left(1, \frac{v_j}{h_j}\right) \\
&\leq \min\left(\sum_{j=1}^d \mathbf{L}_{jj} h_j^2, \sum_{j=1}^d \mathbf{L}_{jj} h_j v_j\right)\|x\|^2 \\
&\leq \min\left(\sum_{j=1}^d \mathbf{L}_{jj} h_j^2, \sqrt{\sum_{j=1}^d \mathbf{L}_{jj}^2 h_j^2}\right)\|x\|^2 = \min\left(\|\mathbf{Diag}(\mathbf{L}) h^2\|_1, \|\mathbf{Diag}(\mathbf{L}) h\|\right)\|x\|^2,
\end{aligned}$$

which implies (15).

E.3. Proof of Theorem 5: DCGD+ with varying quantization steps

Based on the upper bound (15) and the communication constraint given by $\|h_i^{-1}\| = \beta$ for some $\beta > 0$, we get the optimization problem

$$\min_{h_i} \quad \|\mathbf{Diag}(\mathbf{L}_i) h_i\| \quad \text{subject to} \quad \left\|h_i^{-1}\right\| = \beta, \qquad (25)$$

for choosing the optimal quantization parameters $h_{i;j}$. This problem has a closed form solution. Indeed, due to the KKT conditions, we have

$$\frac{\mathbf{L}_{i;j}^2 h_{ij}^4}{\sqrt{\sum_{t=1}^d \mathbf{L}_{i;t}^2 h_{it}^2}} = 2\zeta, \quad \zeta\left(\sum_{t=1}^d h_{ij}^2 - \beta^2\right) = 0,$$

where ζ is the multiplier. Solving this leads to the solution:

$$h_{i;j} = \frac{1}{\beta}\sqrt{\frac{\sum_{t=1}^d \mathbf{L}_{i;t}}{\mathbf{L}_{i;j}}}. \qquad (26)$$

For the solution (26) we have

$$\begin{aligned}
\widetilde{\mathcal{L}}(\mathcal{Q}_{h_i}, \mathbf{L}_i) &\leq \sqrt{\sum_{j=1}^d \mathbf{L}_{i;jj}^2 h_{i;j}^2} = \frac{1}{\beta}\sqrt{\sum_{j=1}^d \mathbf{L}_{i;jj}^2 \frac{\sum_{l=1}^d \mathbf{L}_{i;ll}}{\mathbf{L}_{i;jj}}} = \frac{1}{\beta}\sum_{j=1}^d \mathbf{L}_{i;jj} \\
&\leq \frac{\nu_1}{\beta}\max_{j \in [d]} \mathbf{L}_{i;jj} \leq \frac{\nu_1}{\beta} L_i = \frac{\nu_1}{\beta} L_{\max}. \qquad (27)
\end{aligned}$$

Therefore, if both parameters ν and ν_2 are $\mathcal{O}(1)$, then the rate (6) of DCGD+ becomes $\mathcal{O}(\frac{L_{\max}}{n\mu} + \frac{L_{\max}}{\beta n\mu})$. To make a fair comparison against DCGD, we need to fix $\mathcal{O}(\frac{d}{n})$ number of bits each node communicates to the master server. Now, to make DCGD+ communicate the same number of bits, we set $\beta = \mathcal{O}(\frac{d}{n})$. Hence we have the following iteration complexity for DCGD+ based on solution (26):

$$\mathcal{O}\left(\frac{1}{n}\frac{L_{\max}}{\mu} + \frac{1}{d}\frac{L_{\max}}{\mu}\right)$$

which is $\min(n,d)$ times better than the one of DCGD.

E.4. Proof of Theorem 6: DIANA+ with varying quantization steps

Denote $A_{ij} \overset{\text{def}}{=} \frac{\mathbf{L}_{i;jj}}{n\mu}$. Note that

$$\begin{aligned}
\omega_i + \frac{\mathcal{L}_i}{n\mu} &\le \min\left(\sum_{j=1}^d h_{i;j}^2, \sqrt{\sum_{j=1}^d h_{i;j}^2}\right) + \frac{1}{n\mu}\min\left(\sum_{j=1}^d \mathbf{L}_{i;jj}h_{i;j}^2, \sqrt{\sum_{j=1}^d \mathbf{L}_{i;jj}^2 h_{i;j}^2}\right) \\
&= \min\left(\sum_{j=1}^d h_{i;j}^2, \sqrt{\sum_{j=1}^d h_{i;j}^2}\right) + \min\left(\sum_{j=1}^d \frac{\mathbf{L}_{i;jj}}{n\mu}h_{i;j}^2, \sqrt{\sum_{j=1}^d \left(\frac{\mathbf{L}_{i;jj}}{n\mu}\right)^2 h_{i;j}^2}\right) \\
&\le \min\left(\sum_{j=1}^d h_{i;j}^2 + \sum_{j=1}^d \frac{\mathbf{L}_{i;jj}}{n\mu}h_{i;j}^2, \sqrt{\sum_{j=1}^d h_{i;j}^2} + \sqrt{\sum_{j=1}^d \left(\frac{\mathbf{L}_{i;jj}}{n\mu}\right)^2 h_{i;j}^2}\right) \\
&\le \min\left(\sum_{j=1}^d (1 + A_{ij})\, h_{i;j}^2, \sqrt{2\sum_{j=1}^d \left(1 + A_{ij}^2\right) h_{i;j}^2}\right) \\
&\le \sum_{j=1}^d (1 + A_{ij})\, h_{i;j}^2.
\end{aligned}$$

We solve the optimization problem

$$\min_{h_i} \quad \sum_{j=1}^d (1 + A_{ij})\, h_{i;j}^2 \quad \text{subject to} \quad \left\|h^{-1}\right\| = \beta, \tag{28}$$

which has a closed form solution. Indeed, due to the KKT conditions, we have:

$$h_{i;j} = \frac{1}{\beta}\sqrt{\frac{\sum_{l=1}^d \sqrt{1 + A_{il}^2}}{\sqrt{1 + A_{ij}^2}}}. \tag{29}$$

For the solution (29) we have

$$\begin{aligned}
\omega_i + \frac{\widetilde{\mathcal{L}}_i}{n\mu} &\le \sqrt{2\sum_{j=1}^d \left(1 + A_{ij}^2\right) h_{i;j}^2} = \frac{\sqrt{2}}{\beta}\sum_{j=1}^d \sqrt{1 + A_{ij}^2} = \frac{\sqrt{2}}{\beta}\sum_{j=1}^d (1 + A_{ij}) \\
&= \frac{\sqrt{2}d}{\beta} + \frac{\sqrt{2}}{\beta n\mu}\sum_{j=1}^d \mathbf{L}_{i;jj} \le \frac{\sqrt{2}d}{\beta} + \frac{\sqrt{2}\nu_1}{\beta n}\frac{L_{\max}}{\mu},
\end{aligned}$$

which further leads to $\mathcal{O}(n + \frac{1}{n}\frac{L_{\max}}{\mu} + \frac{1}{d}\frac{L_{\max}}{\mu})$ iteration complexity if $\nu_1 = \mathcal{O}(1)$ and $\beta = \mathcal{O}(\frac{d}{n})$.

F. Notation Table

Table 3. Notation we use throughout the paper.

Basic		
d	number of the model parameters to be trained	
n	number of the nodes/workers in distributed system	
$[n]$	$\overset{\text{def}}{=} \{1, 2, \ldots, n\}$	
$f : \mathbb{R}^d \to \mathbb{R}$	overall empirical loss/risk	(1)
$f_i : \mathbb{R}^d \to \mathbb{R}$	local loss function associated with data owned by the node $i \in [n]$	(1)
$R : \mathbb{R}^d \to \mathbb{R}$	(possibly non-smooth) regularization	(1)
x^*	trained model, i.e. optimal solution to (1)	
ε	target accuracy	
$\lVert x\rVert_0$	$\overset{\text{def}}{=} \#\{j \in [d] \colon x_j \neq 0\}$, number of nonzero entries of $x \in \mathbb{R}^d$	
$\lVert x\rVert$	$\overset{\text{def}}{=} \sqrt{\sum_{j=1}^d x_j^2}$, the standard Euclidean norm of $x \in \mathbb{R}^d$	

Standard		
μ	strong convexity parameter of f	Asm. 2
L	smoothness constant of f, namely $L = \lambda_{\max}(\mathbf{L})$	(2)
L_i	smoothness constant of f_i, namely $L_i = \lambda_{\max}(\mathbf{L}_i)$	
$L_{\max}$	$\overset{\text{def}}{=} \max_{i\in[n]} L_i$	
$\mathcal{C}$	(possibly randomized) compression operator $\mathcal{C} \colon \mathbb{R}^d \to \mathbb{R}^d$	
$\mathbb{B}(\omega)$	class of compressors with $\mathbb{E}\left[\mathcal{C}(x)\right] = x,\ \mathbb{E}\left[\lVert\mathcal{C}(x) - x\rVert^2\right] \leq \omega\lVert x\rVert^2,\ \forall x \in \mathbb{R}^d$	
$\mathcal{C}_i$	compression operator controlled by node i	
ω_i	variance of compression operator $\mathcal{C}_i$	
$\omega_{\max}$	$\overset{\text{def}}{=} \max_{i\in[n]} \omega_i$	
γ	step-size parameter in DCGD+ and DIANA+ methods	
α	learning rate for the local optimal gradients in DIANA+	

Matrix Smoothness		
$\mathbf{L}$	smoothness matrix of f	(3)
$\mathbf{L}^{1/2}$	square root of symmetric and positive semidefinite matrix $\mathbf{L}$	
$\mathbf{L}^{\dagger}$	Moore–Penrose inverse of matrix $\mathbf{L}$	
$\mathbf{L}_i$	smoothness matrix of f_i	
$\mathbf{L}_{i;j}, \mathbf{L}_{i;jj}$	j^{th} diagonal element of $\mathbf{L}_i$	
$\mathcal{L}(\mathcal{C}, \mathbf{L})$	$\overset{\text{def}}{=} \inf\left\{\mathcal{L} \geq 0 \colon \mathbb{E}\lVert\mathcal{C}(x) - x\rVert_{\mathbf{L}}^2 \leq \mathcal{L}\lVert x\rVert^2\ \forall x \in \mathbb{R}^d\right\} \leq \omega\lambda_{\max}(\mathbf{L})$	
$\mathcal{L}_i$	$\overset{\text{def}}{=} \mathcal{L}(\mathcal{C}_i, \mathbf{L}_i)$	(4)
$\mathcal{L}_{\max}$	$\overset{\text{def}}{=} \max_{i\in[n]} \mathcal{L}(\mathcal{C}_i, \mathbf{L}_i) = \max_{i\in[n]} \mathcal{L}_i$	(4)
$\nu,\ \nu_1$	$\nu \overset{\text{def}}{=} \frac{\sum_{i=1}^n L_i}{\max_{i\in[n]} L_i}$ and $\nu_1 \overset{\text{def}}{=} \max_{i\in[n]} \frac{\sum_{j=1}^d \mathbf{L}_{i;j}}{\max_{j\in[d]} \mathbf{L}_{i;j}}$	Def. 8

Quantization		
s	number of quantization levels	
B	number of blocks to divide the space $\mathbb{R}^d$	
l	index for blocks, i.e. $l \in [B]$	
d_l	dimension of the l^{th} subspace in $\mathbb{R}^d$, in particular $\sum_{l=1}^B d_l = d$	
x^l	l^{th} block of coordinates of $x \in \mathbb{R}^d$	
$\mathbf{L}^{ll}$	l^{th} diagonal block matrix of $\mathbf{L}$ with sizes $d_l \times d_l$	
$h_{i;l}$	quantization step of l^{th} block for node i	
β	parameter controlling the number of encoding bits	
j	index for coordinates, i.e. $j \in [d]$	
$h_{i;j}$	quantization step of j^{th} coordinate for node i	(26)