SMOOTHNESS MATRICES BEAT SMOOTHNESS CONSTANTS: BETTER COMMUNICATION COMPRESSION TECHNIQUES FOR DISTRIBUTED OPTIMIZATION

Mher Safaryan
KAUST
Thuwal, Saudi Arabia
mher.safaryan.1@kaust.edu.sa

Filip Hanzely
TTIC
Chicago, USA
filip.hanzely@kaust.edu.sa

Peter Richtárik
KAUST
Thuwal, Saudi Arabia
peter.richtarik@gmail.com

ABSTRACT

Large scale distributed optimization has become the default tool for the training of supervised machine learning models with a large number of parameters and training data. Recent advancements in the field provide several mechanisms for speeding up the training, including *compressed communication*, *variance reduction* and *acceleration*. However, none of these methods is capable of exploiting the inherently rich data-dependent smoothness structure of the local losses beyond standard smoothness constants. In this paper, we argue that when training supervised models, *smoothness matrices*—information-rich generalizations of the ubiquitous smoothness constants—can and should be exploited for further dramatic gains, both in theory and practice. In order to further alleviate the communication burden inherent in distributed optimization, we propose a novel communication sparsification strategy that can take full advantage of the smoothness matrices associated with local losses. To showcase the power of this tool, we describe how our sparsification technique can be adapted to three distributed optimization algorithms—DCGD (Khirirat et al., 2018), DIANA (Mishchenko et al., 2019) and ADIANA (Li et al., 2020)—yielding significant savings in terms of communication complexity. The new methods always outperform the baselines, often dramatically so.

1 INTRODUCTION

In the big data regime, the data is partitioned among many parallel machines, which then cooperatively train a single global model, usually orchestrated by a central server. Distributed training is cast as the distributed optimization problem

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

where d is the number of parameters of model $x \in \mathbb{R}^d$, n is the number of machines participating in the training, $f_i(x)$ is the loss associated with the data stored on machine $i \in [n] \coloneqq \{1, 2, \ldots, n\}$, $f(x)$ is the empirical loss, and $R(x)$ is a regularizer. Ample research over the past two decades has shown that first-order methods are highly scalable and as a result are the methods of choice for distributed optimization problems (Liu & Zhang, 2020). In particular, a substantial amount of work has been devoted to speeding up the training process by developing efficient methods empowered with techniques such as *compressed communication*, *variance reduction* and *acceleration*.

Compressed communication. In distributed training, compute nodes have to communicate with each other, often via a central server, in order to be able to maintain consensus and jointly train a single global model. However, communication of the information pertaining to local progress, which

is typically contained in gradient(s) distilled from local data, is almost invariably *the* key bottleneck in distributed training systems (Xu et al., 2020). One popular way to address this issue is to reduce the number of bits encoding the vector/tensor to be transferred via the help of a lossy *compression operator*. Numerous *unbiased* gradient compression operators have been proposed for this purpose, including sparsifications (Wang et al., 2018; Mishchenko et al., 2020; Alistarh et al., 2018) and quantizations (Alistarh et al., 2017; Zhang et al., 2017; Horváth et al., 2019a; Wu et al., 2018).

Variance reduction. A marked issue that needs to be addressed by successful distributed optimization methods has to do with the (potential) "dissimilarity" of the local loss functions $f_1, \dots, f_n$, which in turn is due to the heterogeneity of the training data defining these functions. The higher the dissimilarity, the harder it is for the devices to find the minimizer of (1). This issue exists even in the unregularized case ($R \equiv 0$). Indeed, while in this case $\frac{1}{n}\sum_i \nabla f_i(x^*) = 0$ if x^* is a minimizer of f, this does not mean that the individual gradients, $\nabla f_1(x^*), \dots, \nabla f_n(x^*)$, are all zero. This issue is exacerbated further by the extra noise coming from gradient compression. Indeed, this noise prevents methods such as Distributed Compressed Gradient Descent (DCGD) (Khirirat et al., 2018) from converging to x^* with a constant learning rate even in the interpolation regime characterized by the identities $\nabla f_i(x^*) = 0$ for all i. Fortunately, these issues can be resolved via carefully designed variance reduction techniques (Gower et al., 2020). In particular, the first variance reduction mechanism for removing the variance coming from compression operators in distributed training is due to Mishchenko et al. (2019), embodied in their DIANA algorithm. The method was initially analyzed for ternary quantization only (Wen et al., 2017), and later generalized to handle a general class of unbiased compression operators (Horváth et al., 2019b; Gorbunov et al., 2020b).

Acceleration. To speed up distributed training even further, it is often possible to employ Nesterov's acceleration technique (Nesterov, 1983; 2004) in concert with gradient compression and variance reduction. For instance, Li et al. (2020) developed the ADIANA method, which adds acceleration on top of a variant of DIANA that relies on the computation of full-batch gradients on all nodes. The resulting method offers provable speedups in convex and strongly convex regimes.

2 Mining for Smoothness Information

One size fits all. Arguably, one of the most ubiquitous, if not *the* most ubiquitous, assumptions used in the literature on first-order optimization methods is that of *L-smoothness* (Nesterov, 2004). A differentiable function $\phi : \mathbb{R}^d \to \mathbb{R}$ is said to be L-smooth if there exists a constant $L \geq 0$ such that for all $x, y \in \mathbb{R}^d$

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

However, most works in the area of finite-sum distributed optimization use it very crudely: they assume that all local loss functions f_i as well as their average, $f = \frac{1}{n}\sum_i f_i$, share the same smoothness constant L (Tang et al., 2019; Woodworth et al., 2020b; Stich, 2020). This is crude because much information is lost this way. Indeed, assuming that each f_i is L_i-smooth, it is well known that f is L_f-smooth with L_f satisfying the inequality $L_f \leq \frac{1}{n}\sum_i L_i$. In the light of this, the above assumption is crude as it effectively replaces the values $L_1, \dots, L_n$ and L_f with a single parameter L satisfying $L \geq \max\{L_1, \dots, L_n\}$. Since the stepsizes and convergence rates of first-order methods depend on the smoothness constant(s) employed, convergence analysis relying on such crude approximation may be significantly suboptimal, and the methods too slow when implemented following the theory.

"Like treasure hidden in a field, which a man found and covered up" (Mat 13:44). The starting point of this paper is the observation that there is a hitherto untapped richness of smoothness information that *can* be used to construct *better distributed optimization algorithms and obtain better theory.* This information is available, but hidden from sight, and is based on the notion of *matrix smoothness*.

Definition 1 (Matrix Smoothness)**.** We say that a differentiable function $\phi : \mathbb{R}^d \to \mathbb{R}$ is $\mathbf{L}$-smooth if there exists a symmetric positive semidefinite matrix $\mathbf{L} \succeq 0$ such that for all $x, y \in \mathbb{R}^d$

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

The standard L-smoothness condition (2) is obtained as a special case of (3) for matrices of the form $\mathbf{L} = L\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Function f_i appearing in (1) is often the average loss over the

training data stored on node i, i.e.,

$$f_i(x) = \frac{1}{m_i} \sum_{m=1}^{m_i} \phi_{im}(\mathbf{A}_{im}x), \tag{4}$$

where $\mathbf{A}_{im} \in \mathbb{R}^{d_{im} \times d}$ is a data matrix, and $\phi_{im} : \mathbb{R}^{d_{im}} \to \mathbb{R}$ is a differentiable function (e.g., the loss over all but the last linear layer of a NN). The following simple result from Qu & Richtárik (2016b), used therein in the context of randomized coordinate descent methods, states that if the loss functions ϕ_{im} are smooth in the standard scalar sense, then f_i is smooth in the matrix sense.

Lemma 1. *If ϕ_{im} is λ_{im}-smooth, then the function f_i defined in* (4) *is $\mathbf{L}_i$-smooth with*

$$\mathbf{L}_i = \frac{1}{m_i} \sum_{m=1}^{m_i} \lambda_{im} \mathbf{A}_{im}^\top \mathbf{A}_{im}. \tag{5}$$

In cases where the local functions f_i are of the form (4)—and it is clear this structure is ubiquitous—there is a lot of potentially useful information contained in the matrix smoothness "constant" $\mathbf{L}_i$. If we were to use the scalar smoothness constant of f_i instead, we would be effectively tossing this richness away, and replacing it with $L_i = \lambda_{\max}(\mathbf{L}_i)$; the largest eigenvalue of $\mathbf{L}_i$. This seems wasteful. As we show in this work, it is. However, we offer a fix.

3 Motivation and Contributions

To the best of our knowledge, *none* of the current distributed optimization methods, including the methods DCGD (Khirirat et al., 2018), DIANA (Mishchenko et al., 2019) and ADIANA (Li et al., 2020) discussed in Section 1, are capable of exploiting the inherently rich data-dependent smoothness structure of the local losses beyond standard smoothness constants. To this effect, we impose the following assumption throughout the paper:

Assumption 1. The functions $f_i \colon \mathbb{R}^d \to \mathbb{R}$ are differentiable, convex, lower bounded[1] and $\mathbf{L}_i$-smooth. Moreover, f is $\mathbf{L}$-smooth with the (standard) smoothness constant $L \coloneqq \lambda_{\max}(\mathbf{L})$.

In this paper, we argue that when training supervised models, *smoothness matrices* (see Definition 1)—information-rich generalizations of the classical and ubiquitous smoothness constants—can and should be exploited for further dramatic gains, both in theory and practice.

3.1. Unbiased diagonal sketches. We study unbiased diagonal sketches, defined as follows:

Definition 2 (Unbiased diagonal sketch)**.** Let S be a random subset of the set of coordinates/features of the model $x \in \mathbb{R}^d$ we wish to train, i.e., $S \subseteq [d] \coloneqq \{1, 2, \ldots, d\}$. Let S be *proper*, i.e., $p_j \coloneqq \mathrm{Prob}(j \in S) > 0$. We now define a random diagonal matrix (sketch) $\mathbf{C} = \mathbf{C}_S \in \mathbb{R}^{d \times d}$ via

$$\mathbf{C} = \mathrm{Diag}(\mathrm{c}_1, \ldots, \mathrm{c}_d), \quad \mathrm{c}_j = \begin{cases} 1/p_j & \text{if } j \in S, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Note that given a vector $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, we have $(\mathbf{C}x)_j = \begin{cases} x_j/p_j & \text{if } \quad j \in S \\ 0 & \text{if } \quad j \notin S \end{cases}$. So, we can control the sparsity level of the product $\mathbf{C}x$ by engineering the properties of the random set S. Also note that $\mathbb{E}[\mathbf{C}x] = x$ for all x.

3.2. Data-dependent sparsification operators. In order to further alleviate the communication burden inherent in distributed optimization, we further propose *data-dependent sparsification operators* that can take full advantage of the smoothness matrices $\mathbf{L}_i$ associated with the local losses f_i. To the best of our knowledge, this is in sharp contrast with the design of all existing tractable compression techniques used in distributed training, which are proposed independently of the training data, and typically based on intuitive or information-theoretic principles. With each node i we associate an unbiased diagonal matrix $\mathbf{C}_i$ of the form (6). We use this and the smoothness matrix of f_i to define a sparsification technique, described next.

Definition 3 (Data-dependent sparsification)**.** In situations when the i-th node wished to communicate local gradient $\nabla f_i(x)$, we ask the node to send the sparse (=compressed) vector $\mathbf{C}_i \mathbf{L}_i^{\dagger 1/2} \nabla f_i(x)$ to

[1] Lower boundedness of $f_i(x)$ can be dropped if $\mathbf{L}_i \succ 0$ is positive definite. This part of the assumption is not a restriction in applications as all loss function are lower bounded.

Table 1: Summary of theoretical results obtained in this work with hidden $\log \frac{1}{\varepsilon}$ factors and constants. Below n is the number of machines, d is the number of parameters of model, $L_{\max} = \max_i L_i$, $L_i = \lambda_{\max}(\mathbf{L}_i)$ and the expected smoothness constant $\tilde{\mathcal{L}}_{\max}$ is defined in (16). The variance of generic compression operator used in the original methods is denoted by ω. In case of sparsification, we have $\omega = {}^{d}/_{\tau} - 1 = \mathcal{O}(n)$ when the expected size of selected coordinates is $\tau = {}^{d}/_{n}$. Parameters ν_1, ν_2 and ν describing distribution of matrices $\mathbf{L}_i$ are defined in (21).

Regime	$\nabla f_i(x^*) \equiv 0$	arbitrary $\nabla f_i(x^*)$	arbitrary $\nabla f_i(x^*)$
Original Methods	**DCGD** (Khirirat et al., 2018)	**DIANA** (Mishchenko et al., 2019)	**ADIANA** (Li et al., 2020)
Iteration Complexity	$\frac{L}{\mu} + \frac{\omega L_{\max}}{n\mu}$	$\omega + \frac{L_{\max}}{\mu} + \frac{\omega L_{\max}}{n\mu}$	$\begin{cases} \omega + \omega\sqrt{\frac{L_{\max}}{n\mu}} & \text{if } n \le \omega \\ \omega + \sqrt{\frac{L_{\max}}{\mu}} + \sqrt{\omega\sqrt{\frac{\omega L_{\max}}{n\mu}}\sqrt{\frac{L_{\max}}{\mu}}} & \text{if } n > \omega \end{cases}$
Iteration Complexity $\tau = {}^{d}/_{n}$	$\frac{L_{\max}}{\mu}$	$n + \frac{L_{\max}}{\mu}$	$n + n\sqrt{\frac{L_{\max}}{n\mu}} \equiv n + \sqrt{n\frac{L_{\max}}{\mu}}$
New Methods	**DCGD+** (Algorithm 1)	**DIANA+** (Algorithm 2)	**ADIANA+** (Algorithm 3)
Iteration Complexity	$\frac{L}{\mu} + \frac{\tilde{\mathcal{L}}_{\max}}{n\mu}$	$\omega_{\max} + \frac{L}{\mu} + \frac{\tilde{\mathcal{L}}_{\max}}{n\mu}$	$\begin{cases} \omega_{\max} + \sqrt{\omega_{\max}\frac{\tilde{\mathcal{L}}_{\max}}{n\mu}} & \text{if } nL \le \tilde{\mathcal{L}}_{\max} \\ \omega_{\max} + \sqrt{\frac{L}{\mu}} + \sqrt{\omega_{\max}\sqrt{\frac{\tilde{\mathcal{L}}_{\max}}{n\mu}}\sqrt{\frac{L}{\mu}}} & \text{if } nL > \tilde{\mathcal{L}}_{\max} \end{cases}$
Iteration Complexity $\tau = {}^{d}/_{n}$	$\frac{L_{\max}}{n\mu} + \frac{L_{\max}}{d\mu}$ (if ν, ν_1 are $\mathcal{O}(1)$)	$n + \frac{L_{\max}}{n\mu} + \frac{L_{\max}}{d\mu}$ (if ν, ν_1 are $\mathcal{O}(1)$)	$\begin{cases} n + n\left(\frac{L_{\max}}{n\mu}\right)^{1/4} & \text{if } nL \le \tilde{\mathcal{L}}_{\max} \\ n + \sqrt{\frac{L_{\max}}{n\mu}} + \left(n\frac{L_{\max}}{\mu}\right)^{3/8} & \text{if } nL > \tilde{\mathcal{L}}_{\max} \end{cases}$ (if ν, ν_2 are $\mathcal{O}(1)$ and $L_{\max}/\mu$ is $\mathcal{O}(nd^2)$)
Reference	Theorem 3, Remark 3	Theorem 4, Remark 4	Theorem 5, Remark 5
Speedup factor (up to)	$\min(n, d)$	$\min(n, d)$	$\begin{cases} \sqrt{d} & \text{if } nL \le \tilde{\mathcal{L}}_{\max} \text{ and } L_{\max}/\mu = \mathcal{O}(nd^2) \\ \sqrt{\min(n,d)} & \text{if } nL > \tilde{\mathcal{L}}_{\max} \text{ and } L_{\max}/\mu = \mathcal{O}(nd^2) \end{cases}$

the server instead. The server then constructs (=decompresses) an unbiased estimator of $\nabla f_i(x)$ as follows:

$$g_i(x) = \mathbf{L}_i^{1/2}\mathbf{C}_i\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x), \tag{7}$$

where $\mathbf{L}_i^{\dagger 1/2}$ denotes the square root of the Moore-Penrose pseudoinverse of $\mathbf{L}_i$.

Notable differences of our proposed communication protocol when compared with standard sparsification techniques are: i) we use the smoothness matrix $\mathbf{L}_i$, ii) the compressed vector $\mathbf{C}_i\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x)$ is not unbiased, iii) we devise a separate decompression mechanism (7), also involving $\mathbf{L}_i$, and this enforces effective unbiasedness.

3.3. Matrix-smoothness-aware redesign of 3 methods. To showcase the power of our approach, we demonstrate how our matrix-smoothness-aware sparsification technique (7) can be adapted to DCGD, DIANA and ADIANA, in each case leading to significant communication savings. By doing so, we show that matrix smoothness can be effectively used to speed up communication compression, variance reduction and acceleration, respectively. This results in three novel methods: DCGD+, DIANA+, and ADIANA+.

3.4. Dramatic improvements in complexity results. We perform complexity analyses for our methods and derive convergence rates under matrix smoothness[2] (see Assumption 1) and strong convexity assumptions (see Theorems 3, 4 and 5). We show that new methods always outperform the originals/baselines, and often dramatically so. Main theoretical results are summarized in Table 1.

REFERENCES

Alekh Agarwal and John C Duchi. Distributed delayed stochastic optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 24, pp. 873–881. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/f0e52b27a7a5d6a1a87373dffa53dbe5-Paper.pdf.

[2]The closest to our result is work of Hanzely & Richtárik (2019b) and their ISEGA method which is able to exploit *diagonal* smoothness matrices. To the best of our knowledge, we are the first to fully exploit smoothness matrices of arbitrary structure, and elevate them as a new tool at the disposal of algorithm designers.

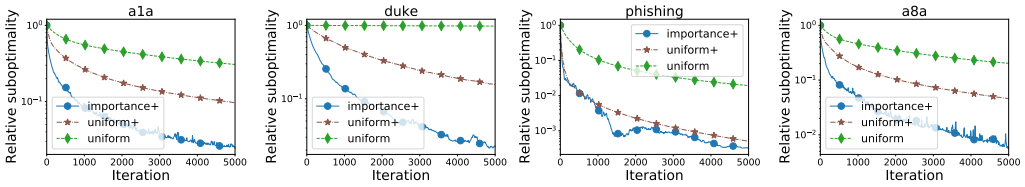

Figure 1: Numerical experiment on logistic regression with LibSVM data Chang & Lin (2011). Comparison of our sparsification strategy of sampling size $\tau = 1$ for DIANA+ (Algorithm 2) using i) importance sampling with probabilities (26), ii) uniform sampling with $p_i = (\frac{1}{d}, \frac{1}{d}, \dots \frac{1}{d})^\top$ and iii) DIANA (Mishchenko et al., 2019) using standard sparsification scheme with uniform sampling. All methods are run with stepsizes as dictated by theory. As expected, this confirms our theoretical findings. First, it demonstrates that our sparsification (7) always outperforms the naive/direct sparsification, sometimes by a large margin. Second, it shows the benefit of importance sampling (26) over the uniform sampling.

Sulaiman Alghunaim, Kun Yuan, and Ali H Sayed. A linearly convergent proximal gradient algorithm for decentralized optimization. In *Advances in Neural Information Processing Systems*, volume 32, pp. 2848–2858. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/e9fd7c2c6623306db59b6aef5c0d5cac-Paper.pdf`.

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 30*, pp. 1709–1720, 2017.

Dan Alistarh, Torsten Hoefler, Mikael Johansson, Nikola Konstantinov, Sarit Khirirat, and Cédric Renggli. The convergence of sparsified gradient methods. In *Neural Information Processing Systems Conf. (NeurIPS)*, 2018.

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *J. Mach. Learn. Res.*, 18(1):8194–8244, January 2017. ISSN 1532-4435.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed optimisation for non-convex problems. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 560–569. PMLR, 2018.

Aleksandr Beznosikov, Samuel Horváth, Peter Richtárik, and Mher Safaryan. On biased compression for distributed learning. *arXiv:2002.12410*, 2020.

Joseph K. Bradley, Aapo Kyrola, Danny Bickson, and Carlos Guestrin. Parallel coordinate descent for L_1-regularized loss minimization. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

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

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

Eduard Gorbunov, Dmitry Kovalev, Dmitry Makarenko, and Peter Richtárik. Linearly converging error compensated sgd. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020b.

Robert M. Gower and Peter Richtárik. Randomized iterative methods for linear systems. *SIAM J. Matrix Anal. Appl.*, 36:1660–1690, 2015.

Robert M. Gower, Mark Schmidt, Francis Bach, and Peter Richtárik. Variance-reduced methods for machine learning. *Proceedings of the IEEE*, 108(11):1968–1983, 2020.

Filip Hanzely and Peter Richtárik. Accelerated coordinate descent with arbitrary sampling and best rates for minibatches. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pp. 304–312. PMLR, 16–18 Apr 2019a. URL http://proceedings.mlr.press/v89/hanzely19a.html.

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

Filip Hanzely, Konstantin Mishchenko, and Peter Richtarik. SEGA: Variance reduction via gradient sketching. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31, pp. 2082–2093. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/fc2c7c47b918d0c2d792a719dfb602ef-Paper.pdf.

Samuel Horváth, Chen-Yu Ho, Ludovít Horváth, Atal Narayan Sahu, Marco Canini, and Peter Richtárik. Natural compression for distributed deep learning. *CoRR*, abs/1905.10988, May 2019a. URL http://arxiv.org/abs/1905.10988.

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

Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian Stich, and Martin Jaggi. Error feedback fixes signSGD and other gradient compression schemes. In *International Conference on Machine Learning*, pp. 3252–3261. PMLR, 2019.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5132–5143. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/karimireddy20a.html.

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

Anastasia Koloskova, Sebastian Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3478–3487. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/koloskova19a.html.

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 Aryeh Kontorovich and Gergely Neu (eds.), *Proceedings of the 31st International Conference on Algorithmic Learning Theory*, volume 117 of *Proceedings of Machine Learning Research*, pp. 451–467, San Diego, California, USA, 08 Feb–11 Feb 2020. PMLR. URL http://proceedings.mlr.press/v117/kovalev20a.html.

Zhize Li, Dmitry Kovalev, Xun Qian, and Peter Richtarik. Acceleration for compressed gradient descent in distributed and federated optimization. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5895–5904. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/li20g.html.

Xiangru Lian, Yijun Huang, Yuncheng Li, and Ji Liu. Asynchronous parallel stochastic gradient for nonconvex optimization. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28, pp. 2737–2745. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/452bf208bf901322968557227b8f6efe-Paper.pdf.

Ji Liu and Ce Zhang. *Distributed Learning Systems with First-Order Methods*, volume 9. Foundations and Trends in Databases, 2020. doi: 10.1561/1900000062.

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

Konstantin Mishchenko, Filip Hanzely, and Peter Richtárik. 99% of worker-master communication in distributed optimization is not needed. In Jonas Peters and David Sontag (eds.), *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pp. 979–988. PMLR, 03–06 Aug 2020.

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

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

Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22:341–362, 2012.

Julie Nutini, Issam Laradji, and Mark Schmidt. Let's make block coordinate descent go fast: Faster greedy rules, message-passing, active-set complexity, and superlinear convergence. *arXiv preprint arXiv:1712.08859*, 2017.

X. Qian, Peter Richtárik, and Tong Zhang. Error compensated distributed SGD can be accelerated. *arXiv: Optimization and Control*, 2020.

Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling II: expected separable overapproximation. *Optimization Methods and Software*, 31:858–884, 2016. doi: 10.1080/10556788.2016.1190361.

Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling I: algorithms and complexity. *Optimization Methods and Software*, 31:829–857, 2016a.

Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling II: algorithms and complexity. *Optimization Methods and Software*, 31:858–884, 2016b.

Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 24, pp. 693–701. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/218a0aefd1d1a4be65601cc6ddc1520e-Paper.pdf.

Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144:1–38, 2014.

Peter Richtárik and Martin Takáč. On optimal probabilities in stochastic coordinate descent methods. *Optim Lett*, 10:1233–1243, 2016a. doi: https://doi.org/10.1007/s11590-015-0916-1.

Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156:433–484, 2016b.

Mher Safaryan and Peter Richtárik. On stochastic sign descent methods. preprint arXiv:1905.12938, 2019.

Mher Safaryan, Egor Shulgin, and Peter Richtárik. Uncertainty principle for communication compression in distributed and federated learning and the search for an optimal compressor. preprint arXiv:2002.08958, 2020.

Amedeo Sapio, Marco Canini, Chen-Yu Ho, Jacob Nelson, Panos Kalnis, Changhoon Kim, Arvind Krishnamurthy, Masoud Moshref, Dan R. K. Ports, and Peter Richtárik. Scaling distributed machine learning with in-network aggregation. In *The 18th USENIX Symposium on Networked Systems Design and Implementation (NSDI '21 Fall)*, 2021. URL http://arxiv.org/abs/1903.06701.

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

Sebastian U. Stich. Local SGD converges fast and communicates little. In *International Conference on Learning Representations*, 2020.

Sebastian U. Stich and Sai Praneeth Karimireddy. The error-feedback framework: Better rates for SGD with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350*, 2019.

Hanlin Tang, Chen Yu, Xiangru Lian, Tong Zhang, and Ji Liu. `DoubleSqueeze`: Parallel stochastic gradient descent with double-pass error-compensated compression. In *Int. Conf. Machine Learning*, volume PMLR 97, pp. 6155–6165, 2019.

Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. PowerSGD: Practical low-rank gradient compression for distributed optimization. In *Neural Information Processing Systems Conf. (NeurIPS)*, 2019.

Hongyi Wang, Scott Sievert, Shengchao Liu, Zachary Charles, Dimitris Papailiopoulos, and Stephen Wright. Atomo: Communication-efficient learning via atomic sparsification. In *Advances in Neural Information Processing Systems*, 2018.

Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Terngrad: Ternary gradients to reduce communication in distributed deep learning. In *Advances in Neural Information Processing Systems*, pp. 1509–1519, 2017.

Blake Woodworth, Kumar Kshitij Patel, Sebastian Stich, Zhen Dai, Brian Bullins, Brendan Mcmahan, Ohad Shamir, and Nathan Srebro. Is local SGD better than minibatch SGD? In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10334–10343. PMLR, 13–18 Jul 2020a. URL `http://proceedings.mlr.press/v119/woodworth20a.html`.

Blake E. Woodworth, Kumar Kshitij Patel, and Nati Srebro. Minibatch vs Local SGD for Heterogeneous Distributed Learning. *Advances in Neural Information Processing Systems 33*, 2020b.

Jiaxiang Wu, Weidong Huang, Junzhou Huang, and Tong Zhang. Error compensated quantized SGD and its applications to large-scale distributed optimization. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5325–5333, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL `http://proceedings.mlr.press/v80/wu18d.html`.

Hang Xu, Chen-Yu Ho, Ahmed M. Abdelmoniem, Aritra Dutta, El Houcine Bergou, Konstantinos Karatsenidis, Marco Canini, and Panos Kalnis. Compressed Communication for Distributed Deep Learning: Survey and Quantitative Evaluation. Technical report, KAUST, Apr 2020. URL `http://hdl.handle.net/10754/662495`.

Hantian Zhang, Jerry Li, Kaan Kara, Dan Alistarh, Ji Liu, and Ce Zhang. 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: "Smoothness Matrices Beat Smoothness Constants: Better Communication Compression Techniques for Distributed Optimization"

CONTENTS

O Variance Reduction with Bi-directional Compression: DIANA++ **53**

A INTRODUCTION

With the desire to build and train high quality machine learning models comes an increased appetite for larger models, both in terms of the number of parameters encoding them, and in the amount of data required to train them. In the big data regime, the data is partitioned among many parallel machines, which then cooperatively train a single global model, usually orchestrated by a central server. Distributed training is cast as the distributed optimization problem

$$\min_{x\in\mathbb{R}^d} f(x) + R(x), \qquad f(x) \coloneqq \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{8}$$

where d is the number of parameters of model $x \in \mathbb{R}^d$, n is the number of machines participating in the training, $f_i(x)$ is the loss associated with the data stored on machine $i \in [n] \coloneqq \{1, 2, \ldots, n\}$, $f(x)$ is the empirical loss, and $R(x)$ is a regularizer. Ample research over the past two decades has shown that first-order methods are highly scalable and as a result are the methods of choice for distributed optimization problems (Liu & Zhang, 2020). In particular, a substantial amount of work has been devoted to speeding up the training process by developing efficient methods empowered with techniques such as *compressed communication*, *variance reduction* and *acceleration*.

A.1 COMPRESSED COMMUNICATION

In distributed training, compute nodes have to communicate with each other, often via a central server, in order to be able to maintain consensus and jointly train a single global model. However, communication of the information pertaining to local progress, which is typically contained in gradient(s) distilled from local data, is almost invariably *the* key bottleneck in distributed training systems (Xu et al., 2020). One popular way to address this issue is to reduce the number of bits encoding the vector/tensor to be transferred via the help of a lossy *compression operator*. Numerous *unbiased* gradient compression operators have been proposed for this purpose, including several types of sparsifications (Wang et al., 2018; Mishchenko et al., 2020; Alistarh et al., 2018) and quantizations (Alistarh et al., 2017; Zhang et al., 2017; Horváth et al., 2019a; Wu et al., 2018). Certain (classes of) *biased* compression operators have been proposed as well, including low-rank approximation (Vogels et al., 2019), sign-based compressors (Seide et al., 2014; Bernstein et al., 2018; Safaryan & Richtárik, 2019) and contractive compressors (Karimireddy et al., 2019; Stich & Karimireddy, 2019; Tang et al., 2019; Beznosikov et al., 2020; Gorbunov et al., 2020b).

A.2 VARIANCE REDUCTION

A marked issue that needs to be addressed by successful distributed optimization methods has to do with the (potential) "dissimilarity" of the local loss functions $f_1, \ldots, f_n$, which in turn is due to the heterogeneity of the training data defining these functions. The higher the dissimilarity, the harder it is for the devices to find the minimizer of (8). This issue exists even in the unregularized case ($R \equiv 0$). Indeed, while in this case $\frac{1}{n}\sum_i \nabla f_i(x^*) = 0$ if x^* is a minimizer of f, this does not mean that the individual gradients, $\nabla f_1(x^*), \ldots, \nabla f_n(x^*)$, are all zero. This shows that local gradient information alone is not enough for any node to "realize" that a solution has been found, which encourages further, in this case unnecessary, iterations. If unaddressed properly, an algorithm is forced to use smaller learning rates, and this leads to unnecessarily slow convergence. On the other hand, when a fixed learning rate is used, the rate is fast, but convergence stops in a potentially large neighborhood[3] of the optimum x^*. This issue is exacerbated further by the extra noise coming from gradient compression. Indeed, this noise prevents methods such as Distributed Compressed Gradient Descent (DCGD) (Khirirat et al., 2018) from converging to x^* with a constant learning rate even in the interpolation regime characterized by the identities $\nabla f_i(x^*) = 0$ for all i. Fortunately, these issues can be resolved via carefully designed variance reduction techniques (Gower et al., 2020). In particular, the first variance reduction mechanism for removing the variance coming from

[3] In the $R \equiv 0$ case, this neighborhood is proportional to the *variance of the local gradients at the optimum*: $\frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(x^*)\|^2$.

compression operators in distributed training is due to Mishchenko et al. (2019), embodied in their DIANA algorithm. The method was initially analyzed for ternary quantization only (Wen et al., 2017), and later generalized to handle a general class of unbiased compression operators (Horváth et al., 2019b; Gorbunov et al., 2020b).

A.3 ACCELERATION

To speed up distributed training even further, it is often possible to employ Nesterov's acceleration technique (Nesterov, 1983; 2004) in concert with gradient compression and variance reduction. For instance, Li et al. (2020) developed the ADIANA method, which adds acceleration on top of a variant of DIANA that relies on the computation of full-batch gradients on all nodes. The resulting method offers provable speedups in convex and strongly convex regimes. Another example is the method ECLK of Qian et al. (2020), which employs compressed communication via any (possibly biased) compressor satisfying a certain contraction property in combination with a slightly different variance reduction technique known as error compensation (Stich & Karimireddy, 2019; Karimireddy et al., 2019), while acceleration is offered by a loopless variant of the accelerated method Katyusha (Allen-Zhu, 2017; Kovalev et al., 2020).

A.4 FURTHER TRICKS

Numerous other techniques are often used to improve some other aspects of distributed training, including implementing multiple local gradient steps before communication (Stich, 2020; Karimireddy et al., 2020; Woodworth et al., 2020a), asynchronous communication protocols (Agarwal & Duchi, 2011; Lian et al., 2015; Recht et al., 2011), in-network aggregation (Sapio et al., 2021), and performing the distributed training in a decentralized peer-to-peer manner without the reliance on an orchestrating server (Koloskova et al., 2019; Alghunaim et al., 2019). However, in this work, we do not explore these directions and focus on the three techniques described before, namely, compressed communication, variance reduction and acceleration.

B MINING FOR SMOOTHNESS INFORMATION

B.1 ONE SIZE FITS ALL

Arguably, one of the most ubiquitous, if not *the* most ubiquitous, assumptions used in the literature on first-order optimization methods is that of L-*smoothness* (Nesterov, 2004). A differentiable function $\phi : \mathbb{R}^d \to \mathbb{R}$ is said to be L-smooth if there exists a constant $L \geq 0$ such that

$$\phi(x) \leq \phi(y) + \langle \nabla \phi(x), x - y \rangle + \frac{L}{2}\|x - y\|^2 \tag{9}$$

holds for all $x, y \in \mathbb{R}^d$. However, most works in the area of finite-sum distributed optimization use it very crudely: they assume that all local loss functions f_i as well as their average, $f = \frac{1}{n}\sum_i f_i$, share the same smoothness constant L (Tang et al., 2019; Woodworth et al., 2020b; Stich, 2020). This is crude because much information is lost this way. Indeed, assuming that each f_i is L_i-smooth, it is well known that f is L_f-smooth with L_f satisfying the inequality $L_f \leq \frac{1}{n}\sum_i L_i$. In the light of this, the above assumption is crude as it effectively replaces the values $L_1, \ldots, L_n$ and L_f with a single parameter L satisfying $L \geq \max\{L_1, \ldots, L_n\}$. Since the stepsizes and convergence rates of first-order methods depend on the smoothness constant(s) employed, convergence analysis relying on such crude approximation may be significantly suboptimal, and the methods too slow when implemented following the theory.

B.2 "ACCORDING TO THE WORK OF THEIR HANDS" (LAM 3:64)

Significant theoretical and practical improvement can often be obtained when taking account of all the smoothness constants involved, avoiding the practice of replacing them all with a single crude bound. Such analyses are more rare, but fairly common. For example, (Richtárik & Takáč, 2016a; Hanzely & Richtárik, 2019a).

B.3 "LIKE TREASURE HIDDEN IN A FIELD, WHICH A MAN FOUND AND COVERED UP" (MAT 13:44)

The starting point of this paper is the observation that there is a hitherto untapped richness of smoothness information that *can* be used to construct *better distributed optimization algorithms and obtain better theory.* This information is available, but hidden from sight, and is based on the notion of *matrix smoothness*.

Definition 4 (Matrix Smoothness)**.** We say that a differentiable function $\phi : \mathbb{R}^d \to \mathbb{R}$ is $\mathbf{L}$-smooth if there exists a symmetric positive semidefinite matrix $\mathbf{L} \succeq 0$ such that

$$\phi(x) \leq \phi(y) + \langle \nabla\phi(y), x - y\rangle + \frac{1}{2}\|x - y\|^2_{\mathbf{L}} \tag{10}$$

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

The standard L-smoothness condition (9) is obtained as a special case of (10) for matrices of the form $\mathbf{L} = L\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Function f_i appearing in (8) is often the average loss over the training data stored on node i, i.e.,

$$f_i(x) = \frac{1}{m_i}\sum_{m=1}^{m_i} \phi_{im}(\mathbf{A}_{im}x), \tag{11}$$

where $\mathbf{A}_{im} \in \mathbb{R}^{d_{im}\times d}$ is a data matrix, and $\phi_{im} : \mathbb{R}^{d_{im}} \to \mathbb{R}$ is a differentiable function (e.g., the loss over all but the last linear layer of a NN). The following simple result from Qu & Richtárik (2016b), used therein in the context of randomized coordinate descent methods, states that if the loss functions ϕ_{im} are smooth in the standard scalar sense, then f_i is smooth in the matrix sense.

Lemma 2. *Assume that each ϕ_{im} is λ_{im}-smooth. Then the function f_i defined in* (11) *is $\mathbf{L}_i$-smooth with*

$$\mathbf{L}_i = \frac{1}{m_i}\sum_{m=1}^{m_i} \lambda_{im}\mathbf{A}_{im}^\top\mathbf{A}_{im}. \tag{12}$$

In cases where the local functions f_i are of the form (11)—and it is clear this structure is ubiquitous—there is a lot of potentially useful information contained in the matrix smoothness "constant" $\mathbf{L}_i$. If we were to use the scalar smoothness constant of f_i instead, we would be effectively tossing this richness away, and replacing it with $L_i = \lambda_{\max}(\mathbf{L}_i)$; the largest eigenvalue of $\mathbf{L}_i$. This seems wasteful. As we show in this work, it is. However, we offer a fix.

C MOTIVATION AND CONTRIBUTIONS

To the best of our knowledge, *none* of the current distributed optimization methods, including the methods DCGD (Khirirat et al., 2018), DIANA (Mishchenko et al., 2019) and ADIANA (Li et al., 2020) discussed in Section A, are capable of exploiting the inherently rich data-dependent smoothness structure of the local losses beyond standard smoothness constants. To this effect, we impose the following assumption throughout the paper:

Assumption 2. The functions $f_i \colon \mathbb{R}^d \to \mathbb{R}$ are differentiable, convex, lower bounded[4] and $\mathbf{L}_i$-smooth. Moreover, f is $\mathbf{L}$-smooth. Let $L \coloneqq \lambda_{\max}(\mathbf{L})$ be the (standard) smoothness constant of f.

In this paper, we argue that when training supervised models, *smoothness matrices* (see Definition 4)—information-rich generalizations of the classical and ubiquitous smoothness constants—can and should be exploited for further dramatic gains, both in theory and practice.

C.1 UNBIASED DIAGONAL SKETCHES

We study unbiased diagonal sketches, defined as follows:

[4]Lower boundedness of $f_i(x)$ can be dropped if $\mathbf{L}_i \succ 0$ is positive definite. This part of the assumption is not a restriction in applications as all loss function are lower bounded.

Table 2: Original and proposed new methods.

ORIGINAL	DCGD	DIANA	ADIANA
NEW	DCGD+ (ALG.1)	DIANA+ (ALG.2)	ADIANA+ (ALG.3)
PROXIMAL	✓	✓	✓
DISTRIBUTED	✓	✓	✓
VARIANCE REDUCED	✗	✓	✓
ACCELERATED	✗	✗	✓

Table 3: Summary of theoretical results obtained in this work with hidden $\log \frac{1}{\varepsilon}$ factors and constants. Below n is the number of machines, d is the number of parameters of model, $L_{\max} = \max_i L_i$, $L_i = \lambda_{\max}(\mathbf{L}_i)$ and the expected smoothness constant $\widetilde{\mathcal{L}}_{\max}$ is defined in (16). The variance of generic compression operator used in the original methods is denoted by ω. In case of sparsification, we have $\omega = {}^{d}/_{\tau} - 1 = \mathcal{O}(n)$ when the expected size of selected coordinates is $\tau = {}^{d}/_{n}$. Parameters ν_1, ν_2 and ν describing distribution of matrices $\mathbf{L}_i$ are defined in (21).

Regime	$\nabla f_i(x^*) \equiv 0$	arbitrary $\nabla f_i(x^*)$	arbitrary $\nabla f_i(x^*)$
Original Methods	**DCGD** (Khirirat et al., 2018)	**DIANA** (Mishchenko et al., 2019)	**ADIANA** (Li et al., 2020)
Iteration Complexity	$\frac{L}{\mu} + \frac{\omega L_{\max}}{n\mu}$	$\omega + \frac{L_{\max}}{\mu} + \frac{\omega L_{\max}}{n\mu}$	$\begin{cases} \omega + \omega\sqrt{\frac{L_{\max}}{n\mu}} & \text{if } n \le \omega \\ \omega + \sqrt{\frac{L_{\max}}{\mu}} + \sqrt{\omega\sqrt{\frac{\omega L_{\max}}{n\mu}}\sqrt{\frac{L_{\max}}{\mu}}} & \text{if } n > \omega \end{cases}$
Iteration Complexity $\tau = {}^{d}/_{n}$	$\frac{L_{\max}}{\mu}$	$n + \frac{L_{\max}}{\mu}$	$n + n\sqrt{\frac{L_{\max}}{n\mu}} \equiv n + \sqrt{n\frac{L_{\max}}{\mu}}$
New Methods	**DCGD+** (Algorithm 1)	**DIANA+** (Algorithm 2)	**ADIANA+** (Algorithm 3)
Iteration Complexity	$\frac{L}{\mu} + \frac{\widetilde{\mathcal{L}}_{\max}}{n\mu}$	$\omega_{\max} + \frac{L}{\mu} + \frac{\widetilde{\mathcal{L}}_{\max}}{n\mu}$	$\begin{cases} \omega_{\max} + \sqrt{\omega_{\max}\frac{\widetilde{\mathcal{L}}_{\max}}{n\mu}} & \text{if } nL \le \widetilde{\mathcal{L}}_{\max} \\ \omega_{\max} + \sqrt{\frac{L}{\mu}} + \sqrt{\omega_{\max}\sqrt{\frac{\widetilde{\mathcal{L}}_{\max}}{n\mu}}\sqrt{\frac{L}{\mu}}} & \text{if } nL > \widetilde{\mathcal{L}}_{\max} \end{cases}$
Iteration Complexity $\tau = {}^{d}/_{n}$	$\frac{L_{\max}}{n\mu} + \frac{L_{\max}}{d\mu}$ (if ν, ν_1 are $\mathcal{O}(1)$)	$n + \frac{L_{\max}}{n\mu} + \frac{L_{\max}}{d\mu}$ (if ν, ν_1 are $\mathcal{O}(1)$)	$\begin{cases} n + n\left(\frac{L_{\max}}{n\mu}\right)^{1/4} & \text{if } nL \le \widetilde{\mathcal{L}}_{\max} \\ n + \sqrt{\frac{L_{\max}}{n\mu}} + \left(n\frac{L_{\max}}{\mu}\right)^{3/8} & \text{if } nL > \widetilde{\mathcal{L}}_{\max} \end{cases}$ (if ν, ν_2 are $\mathcal{O}(1)$ and $L_{\max}/\mu$ is $\mathcal{O}(nd^2)$)
Reference	Theorem 3, Remark 3	Theorem 4, Remark 4	Theorem 5, Remark 5
Speedup factor (up to)	$\min(n, d)$	$\min(n, d)$	$\begin{cases} \sqrt{d} & \text{if } nL \le \widetilde{\mathcal{L}}_{\max} \text{ and } L_{\max}/\mu = \mathcal{O}(nd^2) \\ \sqrt{\min(n,d)} & \text{if } nL > \widetilde{\mathcal{L}}_{\max} \text{ and } L_{\max}/\mu = \mathcal{O}(nd^2) \end{cases}$

Definition 5 (Unbiased diagonal sketch)**.** Let S be a random subset of the set of coordinates/features of the model $x \in \mathbb{R}^d$ we wish to train, i.e., $S \subseteq [d] := \{1, 2, \ldots, d\}$. Let S be *proper*, i.e., $p_j := \text{Prob}(j \in S) > 0$ for all coordinates $j \in [d]$. We now define a random diagonal matrix (sketch) $\mathbf{C} = \mathbf{C}_S \in \mathbb{R}^{d\times d}$ via

$$\mathbf{C} = \text{Diag}(\mathrm{c}_1, \ldots, \mathrm{c}_d), \quad \mathrm{c}_j = \begin{cases} {}^{1}/_{p_j} & \text{if } j \in S, \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

Note that given a vector $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, we have

$$(\mathbf{C}x)_j = \begin{cases} x_j/p_j & \text{if} \quad j \in S \\ 0 & \text{if} \quad j \notin S \end{cases}.$$

So, we can control the sparsity level of the product $\mathbf{C}x$ by engineering the properties of the random set S. Also note that $\mathbb{E}[\mathbf{C}x] = x$ for all x.

C.2 DATA-DEPENDENT SPARSIFICATION OPERATORS

In order to further alleviate the communication burden inherent in distributed optimization, we further propose *data-dependent sparsification operators* that can take full advantage of the smoothness matrices $\mathbf{L}_i$ associated with the local losses f_i. To the best of our knowledge, this is in sharp contrast with the design of all existing tractable compression techniques used in distributed training, which are

proposed independently of the training data, and typically based on intuitive or information-theoretic principles.

With each node i we associate an unbiased diagonal matrix $\mathbf{C}_i$ of the form (13). We use this and the smoothness matrix of f_i to define a sparsification technique, described next.

Definition 6 (Data-dependent sparsification)**.** In situations when the i-th node wished to communicate local gradient $\nabla f_i(x)$, we ask the node to send the sparse (=compressed) vector $\mathbf{C}_i\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x)$ to the server instead. The server then constructs (=decompresses) an unbiased estimator of $\nabla f_i(x)$ as follows:

$$g_i(x) = \mathbf{L}_i^{1/2}\mathbf{C}_i\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x), \tag{14}$$

where $\mathbf{L}_i^{\dagger 1/2}$ denotes the square root of the Moore-Penrose pseudoinverse of $\mathbf{L}_i$.

Notable differences of our proposed communication protocol when compared with standard sparsification techniques are: i) we use the smoothness matrix $\mathbf{L}_i$, ii) the compressed vector $\mathbf{C}_i\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x)$ is not unbiased, iii) we devise a separate decompression mechanism (14), also involving $\mathbf{L}_i$, and this enforces effective unbiasedness.

C.3 Matrix-smoothness-aware redesign of 3 distributed methods

To showcase the power of our approach, we demonstrate how our matrix-smoothness-aware sparsification technique (14) can be adapted to DCGD, DIANA and ADIANA, in each case leading to significant communication savings. By doing so, we show that matrix smoothness can be effectively used to speed up communication compression, variance reduction and acceleration, respectively. This results in three novel methods: DCGD+, DIANA+, and ADIANA+; see Table 2.

C.4 Dramatic improvements in complexity results

We perform complexity analyses for our methods and derive convergence rates under matrix smoothness[5] (see Assumption 10) and strong convexity assumptions (see Theorems 3, 4 and 5). We show that new methods always outperform the originals/baselines, and often dramatically so.

To illustrate the potential of our sparsification technique (14) embedded in the new methods, let all machines $i \in [n]$ use sketches $\mathbf{C}_i$ induced by independent[6] samplings S_i with probabilities $p_{i;j} := \mathrm{Prob}(j \in S_i)$. Then we show that, with optimized probabilities $p_{i;j}$, DCGD+ can be $\mathcal{O}(\min(n,d))$ times faster then DCGD (see Remark 3) and DIANA+ can be $\mathcal{O}(\min(n,d))$ times faster than DIANA (see Remark 4), depending on the distribution of $\mathbf{L}_i$. For the accelerated method, we highlight improvements when condition numbers of subproblems are $\mathcal{O}(nd^2)$. We show that ADIANA+ can be faster than the original ADIANA by a factor of $\mathcal{O}(\sqrt{d})$ in high compression regime, and by a factor of $\mathcal{O}(\sqrt{\min(n,d)})$ in low compression regime (see Remark 5). Main theoretical results are summarized in Table 3.

C.5 Single node case

Specializing our theory to the single machine setting ($n = 1$), we design new non-distributed algorithms providing an alternative viewpoint to randomized coordinate descent methods (see Appendix J).

C.6 Lower bounds

Using matrices as linear compression operators, we further investigate the trade-off between communicated bits and variance induced by the compression (see Appendix K).

[5]The closest to our result is work of Hanzely & Richtárik (2019b) and their ISEGA method which is able to exploit *diagonal* smoothness matrices. To the best of our knowledge, we are the first to fully exploit smoothness matrices of arbitrary structure, and elevate them as a new tool at the disposal of algorithm designers.

[6]Sampling S_i is called independent if $p_{i;jl} := \mathrm{Prob}(\{j,l\} \subseteq S_i) = p_{i;j}p_{i;l}$ for all $j, l \in [d]$.

C.7 EXPERIMENTS

We conduct numerical experiments using LibSVM datasets (Chang & Lin, 2011), confirming the effectiveness and superiority of our sparsification protocol (14) over the standard sparsification scheme (see Section F).

D NEW COMMUNICATION-EFFICIENT DISTRIBUTED METHODS EXPLOITING MATRIX SMOOTHNESS

Consider the distributed optimization problem (8) with the smoothness Assumption 2 and for strongly convex f.

Assumption 3 (μ-convexity)**.** $f\colon \mathbb{R}^d \to \mathbb{R}$ is μ-convex for some $\mu > 0$, i.e.,

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

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

Below we present our new distributed methods, redesigned for matrix smoothness, and their convergence guarantees. Each node $i \in [n]$ generates diagonal sketches $\mathbf{C}_i$ independently from others via an arbitrary sampling S_i and, togther with its smoothness matrix $\mathbf{L}_i$, composes the compression matrix $\mathbf{C}_i\mathbf{L}_i^{\dagger 1/2}$. Probability matrices $\mathbf{P}_i$ and $\widetilde{\mathbf{P}}_i$ associated with the sampling S_i and sketch $\mathbf{C}_i$ are defined as follows

$$\begin{aligned} \mathbf{P}_i &= (p_{i;jl})_{jl=1}^d, \qquad p_{i;jl} = \mathrm{Prob}(\{j,l\} \subseteq S_i), \\ \widetilde{\mathbf{P}}_i &= (\widetilde{p}_{i;jl})_{jl=1}^d, \qquad \widetilde{p}_{i;jl} = \frac{p_{i;jl}}{p_{i;jj}p_{i;ll}} - 1. \end{aligned} \tag{15}$$

Next, we introduce the key quantity, $\widetilde{\mathcal{L}}_{\max}$, describing the joint contribution of our sparsification (14) to the complexities of the three proposed methods:

$$\widetilde{\mathcal{L}}_{\max} = \max_{1\leq i\leq n} \widetilde{\mathcal{L}}_i, \qquad \widetilde{\mathcal{L}}_i = \lambda_{\max}(\widetilde{\mathbf{P}}_i \circ \mathbf{L}_i), \tag{16}$$

Above, $\circ$ stands for Hadamard (i.e. element-wise) product.

D.1 DCGD+

We now present our matrix-smoothness-aware sparsification technique by adapting DCGD algorithm (Khirirat et al., 2018).

Upon receiving the current model x^k from the server, each node computes $\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^k)$ based on local training data and smoothness matrix. Next, sparsified updates $\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^k)$ are sent back to the server, which then averages decompressed updates $\mathbf{L}_i^{1/2}\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^k)$ and performs proximal step to get a new model x^{k+1}.

Algorithm 1 DCGD+

Input: Initial point $x^0 \in \mathbb{R}^d$, current point x^k, step size γ, diagonal sketch $\mathbf{C}_i^k$
on server
  send x^k to all nodes
  get sparse updates $\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^k)$ from each node
  $g^k = \frac{1}{n}\sum_{i=1}^n \mathbf{L}_i^{1/2}\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^k)$
  $x^{k+1} = \mathrm{prox}_{\gamma R}(x^k - \gamma g^k)$

With this method we get convergence up to a neighborhood.

Theorem 3 (see L.3). *Let Assumptions 2 and 3 hold and assume that each node generates its own diagonal sketch $\mathbf{C}_i$ independently from others. Then, for the step-size*

$$0 < \gamma \leq \frac{1}{L + \frac{2}{n}\widetilde{\mathcal{L}}_{\max}},$$

the iterates $\{x^k\}$ of Algorithm 1 satisfy

$$\mathbb{E}\left[\|x^k - x^*\|^2\right] \leq (1-\gamma\mu)^k \|x^0 - x^*\|^2 + \frac{2\gamma\sigma^*}{\mu n}, \tag{17}$$

where $\sigma^* \coloneqq \frac{1}{n}\sum_{i=1}^n \widetilde{\mathcal{L}}_i \|\nabla f_i(x^*)\|^2_{\mathbf{L}_i^\dagger}$.

Proof technique. First we show the unbiasedness of g^k. As smoothness matrices $\mathbf{L}_i$ are not necessarily invertible, terms like $\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}$ show up in the analysis and block chains of cancellations. This part is handled by the fact that gradients $\nabla f_i(x)$ of an $\mathbf{L}_i$-smooth function are constraint to remain in $\operatorname{Range}\mathbf{L}_i$ and the mapping associated with the matrix $\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}$ is identity on the subspace $\operatorname{Range}(\mathbf{L}_i)$. Second part is the tight estimation of $\mathbb{E}_k\|g^k - \nabla f(x^*)\|^2$, which describes the progress of the method in the presence of stochasticity. Key part is getting the decomposition

$$\mathbb{E}_k\left[\|g^k - \nabla f(x^*)\|^2\right] = \|\nabla f(x^k) - \nabla f(x^*)\|^2 + \frac{1}{n^2}\sum_{i=1}^n \left\|\nabla f_i(x^k)\right\|^2_{\mathbf{L}_i^{\dagger 1/2}(\widetilde{\mathbf{P}}_i \circ \mathbf{L}_i)\mathbf{L}_i^{\dagger 1/2}}, \tag{18}$$

which shows the exact interaction between random sketches and local smoothness. We complete the proof using the unified convergence theory of Gorbunov et al. (2020a).

D.2 Variance reduction: DIANA+

Next, we apply our sparsification technique to the variance reduced method DIANA (Mishchenko et al., 2019).

In this method, each node maintains an auxiliary control vector h_i^k, called shift, which helps to reduce the variance coming from the sparsification. Moreover, the central server keeps track of only the averaged shift h^k. Then, the model x^k as well as control vectors $h_i^k,\ h^k$ are updated by decompressing sparse information Δ_i^k using matrices $\mathbf{L}_i$.

Algorithm 2 DIANA+

Input: Initial point $x^0 \in \mathbb{R}^d$, initial shifts $h_i^0 \in \operatorname{Range}(\mathbf{L}_i)$, current point x^k, step size parameter γ and α, sketch $\mathbf{C}_i^k$ and $\overline{\mathbf{C}}_i^k \coloneqq \mathbf{L}_i^{1/2}\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}$, current shifts $h_1^k, \ldots, h_n^k$ and $h^k \coloneqq \frac{1}{n}\sum_{i=1}^n h_i^k$.
on each node
  get x^k from the server
  send sparse update $\Delta_i^k = \mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x^k) - h_i^k)$
  $\overline{\Delta}_i^k = \mathbf{L}_i^{1/2}\Delta_i^k,\ g_i^k = h_i^k + \overline{\Delta}_i^k, h_i^{k+1} = h_i^k + \alpha\overline{\Delta}_i^k$
on server
  get sparse updates Δ_i^k from each node
  $\overline{\Delta}^k = \frac{1}{n}\sum_{i=1}^n \overline{\Delta}_i^k = \frac{1}{n}\sum_{i=1}^n \mathbf{L}_i^{1/2}\Delta_i^k$
  $g^k = \overline{\Delta}^k + h^k = \frac{1}{n}\sum_{i=1}^n \overline{\mathbf{C}}_i^k\left(\nabla f_i(x^k) - h_i^k\right) + h_i^k$
  $x^{k+1} = \operatorname{prox}_{\gamma R}(x^k - \gamma g^k)$
  $h^{k+1} = h^k + \alpha\overline{\Delta}^k$

In this case we get rid of the neighborhood and provide linear convergence to the exact solution x^*. We use $\widetilde{\mathcal{O}}$ notation to ignore $\log\frac{1}{\varepsilon}$ factors and constants.

Theorem 4 (see L.4). *Let Assumptions 2 and 3 hold and assume that each node generates its own diagonal sketch $\mathbf{C}_i$ independently from others. Then, for the step-size*

$$\gamma = \frac{1}{L + \frac{6}{n}\widetilde{\mathcal{L}}_{\max}},$$

Algorithm 2 guarantees $\mathbb{E}\left[\|x^k - x^*\|^2\right] \leq \varepsilon$ *after*

$$\widetilde{\mathcal{O}}\left(\omega_{\max} + \frac{L}{\mu} + \frac{\widetilde{\mathcal{L}}_{\max}}{n\mu}\right) \tag{19}$$

iterations, where $\omega_{\max} = \max_{1\leq i\leq n}\omega_i$ *and* $\omega_i = \max_{1\leq j\leq d}\frac{1}{p_{i;j}} - 1$ *is the variance of compression operator induced by sketch* $\mathbf{C}_i$.

Proof technique. The structure of the proof resembles the one for DCGD+. With the introduced shift vectors, the unbiasedness of g^k additionally requires $h_i^k \in \mathrm{Range}(\mathbf{L}_i)$. This is resolved by the initialization $h_i^0 \in \mathrm{Range}(\mathbf{L}_i)$ and linear update rule for h_i^{k+1} in line 5. The proof develops a decomposition similar to (18) with modified second term $\sigma^k \coloneqq \frac{1}{n}\sum_{i=1}^n \|h_i^k - \nabla f(x^*)\|^2_{\mathbf{L}_i^\dagger}$ involving shifts h_i^k. To avoid the neighborhood term in (17) and guarantee a linear convergence for x^k, we make σ^k converge linearly too. Key technical part of the proof is to establish contracting recurrence relation for σ^k which boils down to $\mathbb{E}[\overline{\mathbf{C}}_i^\top \mathbf{L}_i^\dagger \overline{\mathbf{C}}_i] \preceq (\omega_i + 1)\mathbf{L}_i^\dagger$. The latter bound justifies the structure of $\overline{\mathbf{C}}_i$ as it filters the interaction between compression and smoothness mixed in the expectation and separates variance ω_i of compression from smoothness matrix $\mathbf{L}_i$.

Remark 1 (Variance Reduction: ISEGA+)**.** *In Appendix N we apply our redesign to another variance reduced method called ISEGA (Mishchenko et al., 2020; Hanzely & Richtárik, 2019b). At the core of ISEGA, the mechanism for variance reduction is based on SEGA method (Hanzely et al., 2018). The key difference between ISEGA and DIANA is that ISEGA updates the control variates* h *more aggressively using projection instead of the mere* α*-step towards the projection used in DIANA. Formally, adapting our matrix-smoothness-aware sparsification to ISEGA, we define the update rule of control vectors* h_i^k *as follows*

$$h_i^{k+1} = \operatorname*{arg\,min}_{\substack{h\in\mathrm{Range}(\mathbf{L}_i)\\ \mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^k)=\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}h}} \|h - h_i^k\|^2_{\mathbf{L}_i^\dagger} = h_i^k + \mathbf{L}_i^{1/2}\mathbf{Diag}(\mathbf{P}_i)\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x^k) - h_i^k).$$

On the other hand, notice that the update rule in DIANA+ has the form

$$h_i^{k+1} = h_i^k + \alpha\mathbf{L}_i^{1/2}\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x^k) - h_i^k)$$

for some fixed scalar $\alpha > 0$*, and thus is more conservative. Note that we choose the gradient estimator for ISEGA+ to be the same* $g_i^k = h_i^k + \mathbf{L}_i^{1/2}\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x^k) - h_i^k)$*. The method is presented as Algorithm 7 in Appendix N.*

In contrast to DIANA+, we can not obtain the convergence rate of ISEGA+ directly from the framework of Gorbunov et al. (2020a). Instead, to get the tight convergence rate, we shall cast it as an instance of GJS method (Hanzely & Richtárik, 2019b). Theorem 23 provides the result – we can see that the worst case complexity is identical to DIANA+. However, in terms of the practical performance, we expect ISEGA+ to outperform DIANA+ due to the more aggressive update rule of control variates.

Remark 2 (Variance Reduction with Bi-directional Compression: DIANA++)**.** *As an extension to DIANA+, in Appendix O we apply our sparsification technique both for nodes and for the central server, thus compressing gradients in both directions of communication. We develop and analyze DIANA++ method (see Algorithm 8), for which the central server applies compression in its turn with sketch* $\mathbf{C}$ *independently. To converge in a linear rate, DIANA++ maintains an additional control vector, which helps to reduce the variance coming from the master's sparsification. Theorem 24 provides complexity result for DIANA++, which recovers the same complexity (19) of DIANA+ if no compression is applied by the master.*

D.3 Acceleration with variance reduction: ADIANA+

Finally, we redesign the accelerated method ADIANA (Li et al., 2020) to effectively exploit local smoothness matrices.

The algorithm develops four sequences $\{x^k, y^k, z^k, w^k\}$ of models, which are layered via convex combinations, proximal steps and probabilistic assignments. In each iteration, nodes receive models

x^k and w^k from the server, and send back sparse updates Δ_i^k and δ_i^k using local data and control vectors h_i^k. Then, decompressing these sparse vectors with matrices $\mathbf{L}_i$, nodes update their shifts h_i^k and the server updates all four models along with averaged shift h^k.

Algorithm 3 ADIANA+

Input: Initial points $x^0 = y^0 = z^0 = w^0 \in \mathbb{R}^d$, initial shifts $h_i^0 \in \mathrm{Range}(\mathbf{L}_i)$, current point x^k, parameters $\gamma, \alpha, \beta, \eta, \theta_1, \theta_2, q$, sketch $\mathbf{C}_i^k$ and $\overline{\mathbf{C}}_i^k := \mathbf{L}_i^{1/2}\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}$, current shifts $h_1^k, \cdots, h_n^k$ and $h^k = \frac{1}{n}\sum_{i=1}^n h_i^k$
on server
  $x^k = \theta_1 z^k + \theta_2 w_k + (1-\theta_1-\theta_2)y^k$
  send x^k and w^k to all nodes
on each node
  send sparse update $\Delta_i^k = \mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x^k) - h_i^k)$
  send sparse update $\delta_i^k = \mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(w^k) - h_i^k)$
  update local gradient $\overline{\Delta}_i^k = \mathbf{L}_i^{1/2}\Delta_i^k,\ g_i^k = h_i^k + \overline{\Delta}_i^k$
  update local shift $\bar{\delta}_i^k = \mathbf{L}_i^{1/2}\delta_i^k,\ h_i^{k+1} = h_i^k + \alpha\bar{\delta}_i^k$
on server
  get sparse updates Δ_i^k and δ_i^k from each node
  $\overline{\Delta}^k = \frac{1}{n}\sum_{i=1}^n \mathbf{L}_i^{1/2}\Delta_i^k,\ \bar{\delta}^k = \frac{1}{n}\sum_{i=1}^n \mathbf{L}_i^{1/2}\delta_i^k$
  $g^k = \overline{\Delta}^k + h^k = \frac{1}{n}\sum_{i=1}^n \overline{\mathbf{C}}_i^k\left(\nabla f_i(x^k) - h_i^k\right) + h_i^k$
  $h^{k+1} = h^k + \alpha\bar{\delta}^k$
  $y^{k+1} = \mathrm{prox}_{\eta R}(x^k - \eta g^k)$
  $z^{k+1} = \beta z^k + (1-\beta)x^k + \frac{\gamma}{\eta}(y^{k+1} - x^k)$
  $w^{k+1} = \begin{cases} y^k & \text{with probability} \quad q, \\ w^k & \text{with probability} \quad 1-q. \end{cases}$

Clearly, the new method ADIANA+ enjoys the accelerated rate, which is strictly better then the one for DIANA+.

Theorem 5 (see L.5)**.** *Let Assumptions 2 and 3 hold and assume that each node generates its own diagonal sketch $\mathbf{C}_i$ independently from others. Then, the iteration complexity of Algorithm 3 guaranteeing $\mathbb{E}\left[\|z^k - x^*\|^2\right] \leq \varepsilon$ is*

$$\begin{cases} \widetilde{\mathcal{O}}\left(\omega_{\max} + \sqrt{\omega_{\max}\frac{\widetilde{\mathcal{L}}_{\max}}{\mu n}}\right) & \text{if} \quad nL \leq \widetilde{\mathcal{L}}_{\max} \\ \widetilde{\mathcal{O}}\left(\omega_{\max} + \sqrt{\frac{L}{\mu}} + \sqrt{\omega_{\max}\sqrt{\frac{\widetilde{\mathcal{L}}_{\max}}{\mu n}}\sqrt{\frac{L}{\mu}}}\right) & \text{if} \quad nL > \widetilde{\mathcal{L}}_{\max}. \end{cases} \tag{20}$$

Proof technique. The additional difficulty that acceleration brings on top of variance reduction is the modified term $H^k := \frac{1}{n}\sum_{i=1}^n \|h_i^k - \nabla f_i(w^k)\|^2_{\mathbf{L}_i^\dagger}$ controlling variance reduction process. The subtlety of H^k in contrast to σ^k is gradients $\nabla f_i(w^k)$ which are not fixed. Key technical part is to reduce contracting property of H^k into upper bounding $\mathbb{E}[(\mathbf{I} - \alpha\overline{\mathbf{C}}_i)^\top\mathbf{L}_i^\dagger(\mathbf{I} - \alpha\overline{\mathbf{C}}_i)]$ by $(1-\alpha)\mathbf{L}_i^\dagger$ as quadratic forms in the subspace $\mathrm{Range}(\mathbf{L}_i)$.

E Improvements Over the Original Methods

To compare the proposed methods with originals and highlight improvement factors, we choose independent sampling for all nodes. For Algorithms 1 and 2, we optimize probabilities of the samplings based on the complexities we found.

E.1 PARAMETERS DESCRIBING DISTRIBUTION OF $\mathbf{L}_i$

Define parameters ν and ν_s describing the distribution of local smoothness matrices $\mathbf{L}_i$ as follows

$$\nu \coloneqq \frac{\sum_{i=1}^n L_i}{\max_{i\in[n]} L_i}, \quad \nu_s \coloneqq \max_{i\in[n]} \frac{\sum_{j=1}^d \mathbf{L}_{i;j}^{1/s}}{\max_{j\in[d]} \mathbf{L}_{i;j}^{1/s}}, \tag{21}$$

where $L_i = \lambda_{\max}(\mathbf{L}_i)$ and $s = 1$ or $s = 2$. Let $L_{\max} \coloneqq \max_{1\leq i\leq n} L_i$. Note that parameters $\nu \in [1, n]$ and $\nu_s \in [1, d]$ describe the distribution over the nodes and coordinates respectively. If $\mathbf{L}_i$ are distributed uniformly, then $\nu = n$ and $\nu_s = d$. On the other extreme, when the distribution is extremely non-uniform, we have $\nu \ll n$ and $\nu_s \ll d$. These parameters are used to highlight the range of iteration complexities new methods can provide.

E.2 IMPORTANCE SAMPLING FOR DCGD+

Let $\tau = \mathbb{E}\left[|S_i|\right] = \sum_{j=1}^d p_{i;j}$ be the expected mini-batch size for the samplings S_i, where $p_{i;j} = p_{i;jj}$. Notice that convergence rate of Algorithm 1 depends on $\widetilde{\mathcal{L}}_{\max} = \max_{1\leq i\leq n} \widetilde{\mathcal{L}}_i$. Since each node $i \in [n]$ generates its own diagonal sketch $\mathbf{C}_i$ independently from others, each node can optimize $\widetilde{\mathcal{L}}_i = \lambda_{\max}(\widetilde{\mathbf{P}}_i \circ \mathbf{L}_i)$ independently based on local smoothness matrix $\mathbf{L}_i$. In general, minimizing $\lambda_{\max}(\widetilde{\mathbf{P}}_i \circ \mathbf{L}_i)$ with respect to probability matrix $\widetilde{\mathbf{P}}_i$ is hard. However, when each node uses an independent sampling, which means $p_{i;jl} = p_{i;j}p_{i;l}$ if $j \neq l$, then

$$\lambda_{\max}(\widetilde{\mathbf{P}}_i \circ \mathbf{L}_i) = \max_{1\leq j\leq d} \left(\frac{1}{p_{i;j}} - 1\right) \mathbf{L}_{i;j}, \tag{22}$$

for which we can find the optimal probabilities $p_{i;j}$. To minimize the maximum term in (22), we should have $\left({}^1/_{p_{i;j}} - 1\right) \mathbf{L}_{i;j} = \rho_i$ for some $\rho_i \geq 0$. Then the solution is

$$p_{i;j} = \frac{\mathbf{L}_{i;j}}{\mathbf{L}_{i;j} + \rho_i}, \tag{23}$$

where $\rho_i \geq 0$ is the unique solution to $\sum_{j=1}^d \frac{\mathbf{L}_{i;j}}{\mathbf{L}_{i;j}+\rho_i} = \tau$. The latter does not allow closed form solution for ρ_i. However, since ρ_i is the root of strictly monotone and one dimensional function, it can be computed numerically using one dimensional solvers. Thus, we can efficiently compute the optimal probabilities (23).

Proposition 6 (Optimality)**.** *The independent sampling with probabilities (23) is the optimal independent sampling for the rate (17).*

Remark 3 (Improvement over DCGD (Khirirat et al., 2018))**.** *With probabilities (23) we show in Appendix M.1 that*

$$\frac{L}{\mu} + \frac{\widetilde{\mathcal{L}}_{\max}}{n\mu} \leq \left(\frac{\nu}{n} + \frac{\nu_1}{\tau n}\right) \frac{L_{\max}}{\mu}. \tag{24}$$

In the interpolation regime (i.e. $\nabla f_i(x^*) = 0$ for all $i \in [n]$), the iteration complexity of DCGD is $\widetilde{\mathcal{O}}(\frac{L}{\mu} + \frac{\omega L_{\max}}{n\mu})$ for general compression operator with variance parameter ω. If we specialize compression to sparsification with $\tau = {}^d/_n$ entries (which gives $\omega = {}^d/_\tau - 1 = n - 1$), we get $\widetilde{\mathcal{O}}(\frac{L_{\max}}{\mu})$. Notice that, in this regime, Theorem 3 also provides linear convergence with iteration complexity $\widetilde{\mathcal{O}}(\frac{L}{\mu} + \frac{\widetilde{\mathcal{L}}_{\max}}{n\mu})$. Based on (24), it is bounded by $\widetilde{\mathcal{O}}((\frac{\nu}{n} + \frac{\nu_1}{d})\frac{L_{\max}}{\mu})$, which is always better than $\widetilde{\mathcal{O}}(\frac{L_{\max}}{\mu})$ and can be as small as $\widetilde{\mathcal{O}}(\frac{L_{\max}}{\min(n,d)\mu})$. Hence, for mini-batch $\tau = {}^d/_n$, DCGD+ (Algorithm 1) guarantees the same $\widetilde{\mathcal{O}}(\frac{L_{\max}}{\mu})$ complexity in the worst case, but could provide up to $\min(n, d)$ times speedup.

E.3 IMPORTANCE SAMPLING FOR DIANA+

To find optimal probabilities for DIANA+, we minimize $\omega_{\max} + \frac{\widetilde{\mathcal{L}}_{\max}}{\mu n}$ part of the complexity (19). Definitions of $\widetilde{\mathcal{L}}_{\max}$ and $\omega_{\max}$ imply that it is equivalent to minimize

$$\max_{1\leq j\leq d} \left(\frac{1}{p_{i;j}} - 1\right) \mathbf{L}'_{i;j}, \quad \mathbf{L}'_{i;j} \coloneqq \frac{\mathbf{L}_{i;j}}{\mu n} + 1, \tag{25}$$

which can be solved in the same way as (22) yielding

$$p_{i;j} = \frac{\mathbf{L}'_{i;j}}{\mathbf{L}'_{i;j} + \rho'_i} = \frac{\mathbf{L}_{i;j} + \mu n}{\mathbf{L}_{i;j} + (1 + \rho'_i)\mu n}. \tag{26}$$

Proposition 7 (Optimality)**.** *The independent sampling with probabilities (26) is the optimal[7] independent sampling for the complexity (19).*

Remark 4 (Improvement over DIANA (Mishchenko et al., 2019; Horváth et al., 2019b))**.** *Here we compare DIANA+ against the original DIANA method, which has iteration complexity $\widetilde{\mathcal{O}}(n + \frac{L_{\max}}{\mu})$ when each node sparsifies with $\tau = {}^{d}/_{n}$ entries. With probabilities (26) we upper bound the complexity (19) in Appendix M.2 as follows*

$$\omega_{\max} + \frac{L}{\mu} + \frac{\widetilde{\mathcal{L}}_{\max}}{\mu n} \leq \frac{2d}{\tau} + \left(\frac{\nu}{n} + \frac{2\nu_1}{\tau n}\right)\frac{L_{\max}}{\mu}. \tag{27}$$

Therefore, with $\tau = {}^{d}/_{n}$, DIANA+ (Algorithm 2) guarantees the same $\widetilde{\mathcal{O}}(n + \frac{L_{\max}}{\mu})$ complexity in the worst case, but could provide up to $\min(n, d)$ times speedup with iteration complexity $\widetilde{\mathcal{O}}(n + \frac{L_{\max}}{\min(n,d)\mu})$.

E.4 Independent sampling for ADIANA+

Clearly, if we sparsify with uniform probabilities $p_{i;j} = {}^{\tau}/_{d}$, then Algorithm 3 recovers the rate of ADIANA.

Remark 5 (Improvement over ADIANA (Li et al., 2020))**.** *To show that the rate could be significantly better in some cases, consider the following choice*

$$p_{i;j} = \sqrt{\frac{\mathbf{L}'_{i;j}}{\mathbf{L}'_{i;j} + \rho''_i}}, \quad \mathbf{L}'_{i;j} = \frac{\mathbf{L}_{i;j}}{\mu n} + 1, \tag{28}$$

where ρ''_i is determined uniquely from $\sum_{j=1}^{d} p_{i;j} = \tau$. Then, with these probabilities and for ${}^{L_{\max}}/_{\mu} = \mathcal{O}(nd^2)$, we show in Appendix M.3 that

$$\frac{L}{\mu} \leq \frac{\nu L_{\max}}{n\mu}, \quad \omega_{\max} = \mathcal{O}\left(\frac{\nu_2 d}{\tau}\right), \quad \frac{\mathcal{L}_{\max}}{\mu n} = \mathcal{O}\left(\frac{\nu_2 d}{\tau}\sqrt{\frac{L_{\max}}{n\mu}}\right).$$

Furthermore, assuming both ν and ν_2 are $\mathcal{O}(1)$, choosing $\tau = {}^{d}/_{n}$ we get

$$\frac{L}{\mu} \leq \mathcal{O}\left(\frac{L_{\max}}{n\mu}\right), \quad \omega_{\max} = \mathcal{O}\left(n\right), \quad \frac{\mathcal{L}_{\max}}{\mu n} = \mathcal{O}\left(\sqrt{\frac{nL_{\max}}{\mu}}\right).$$

Then, the complexity (20) of ADIANA+ reduces to

$$\begin{cases} n + n\left(\frac{L_{\max}}{n\mu}\right)^{1/4} & \text{if } nL \leq \widetilde{\mathcal{L}}_{\max}, \\ n + \sqrt{\frac{L_{\max}}{n\mu}} + \left(n\frac{L_{\max}}{\mu}\right)^{3/8} & \text{if } nL > \widetilde{\mathcal{L}}_{\max}, \end{cases}$$

which, compared to the complexity of ADIANA with $\omega = \mathcal{O}(n)$ compression, gives $\sqrt{d}$ times improvement in the first case and $\sqrt{\min(n, d)}$ times improvement in the second case (ignoring the first summand n of the complexities).

F Experiments

In this section we numerically compare the proposed matrix-smoothness-aware sparsification strategy (14) with the usual sparsification scheme.

[7] In the sense that it minimizes a quantity, which is the complexity of DIANA+ up to some constant factor.

F.1 EXPERIMENTAL SETUP

We devise three different experiments on logistic regression with LibSVM data (Chang & Lin, 2011). In particular, the objective is given as

$$f_i(x) \coloneqq \frac{1}{m_i}\sum_{j=1}^{m_i}\log\left(1+\exp\left((\mathbf{A}_{im})_{j,:}x\cdot(b_{im})_j\right)\right)+\frac{\mu}{2}\|x\|^2,$$

where $\mathbf{A}_{im}\in\mathbb{R}^{d_{im}\times d}$ is the data matrix with corresponding labels $b_{im}\in\mathbb{R}^{d_{im}}$. In our case, we did split the randomly reshuffled datasets into equal chunks among workers in each case so that $m_i = m_j$ for all $i, j \leq n$. The data matrix $\mathbf{A}$ was normalized so that each datapoint has a norm equal to $\frac{1}{2}$. Lastly, we have chosen $\mu = 10^{-3}$ for all experiments.

For each of the datasets, we have selected a specific number of workers given by Table 4. Each of the method was run with theory supported parameters with an exception of the ADIANA+, where we have omitted several constant factors for the sake of practicality.

Table 4: Datasets.

Dataset	# datapoints	d	n	m_i
a1a	1 605	123	107	15
mushrooms	8 124	112	12	677
phishing	11 055	68	11	1 005
madelon	2 000	500	4	500
duke	44	7 129	4	11
a8a	22 696	123	8	2837

F.2 VARIANCE REDUCTION WITH NEW SPARSIFICATION AND IMPORTANCE SAMPLING

We now comment on the experiment illustrated in Figure 2. We examine three sparsification schemes (two variants of our strategy and the usual sparsification not aware of smoothness matrices) and their influence on convergence using six different datasets. Considered schemes are i) DIANA+ with importance sampling (26), ii) DIANA+ with uniform sampling, and iii) DIANA with uniform sampling, i.e., uniform sparsification unaware of smoothness matrices. In all three cases we fixed the sampling size $\tau = 1$.

As expected, Figure 2 confirms our theoretical findings. First, it demonstrates that our sparsification (14) always outperforms the naive/direct sparsification, sometimes by a large margin. Second, it shows the benefit of importance sampling (26) over the uniform sampling.

F.3 THE PROPOSED AND USUAL SPARSIFICATION TECHNIQUES FOR THE 3 DISTRIBUTED METHODS

In the second experiment depicted in Figure 3, we compare six different methods: well-established DCGD, DIANA, ADIANA and our methods DCGD+, DIANA+, ADIANA+, all with uniform sampling for $\tau = 1$. In order to highlight the importance of the variance reduction, in this experiment we choose the starting point to be close to the optimum.

Figure 3 demonstrates the following: i) methods with matrix-aware sparsification (i.e., DCGD+, DIANA+, ADIANA+) always outperform their baselines (i.e., DCGD, DIANA, ADIANA) ii) acceleration almost always outperforms the non-accelerated variant, often dramatically so and iii) variance reduction never hurts the convergence, but often stabilizes the oscillation of the non-variance reduced counterpart.

F.4 THE EFFECT OF SPARSIFICATION LEVEL τ ON THE CONVERGENCE RATE

In this experiment, we study the effect of sparsification level τ on the convergence rate. Informally speaking, our theory suggests that the sparsification does not hurt the convergence rate unless τ is

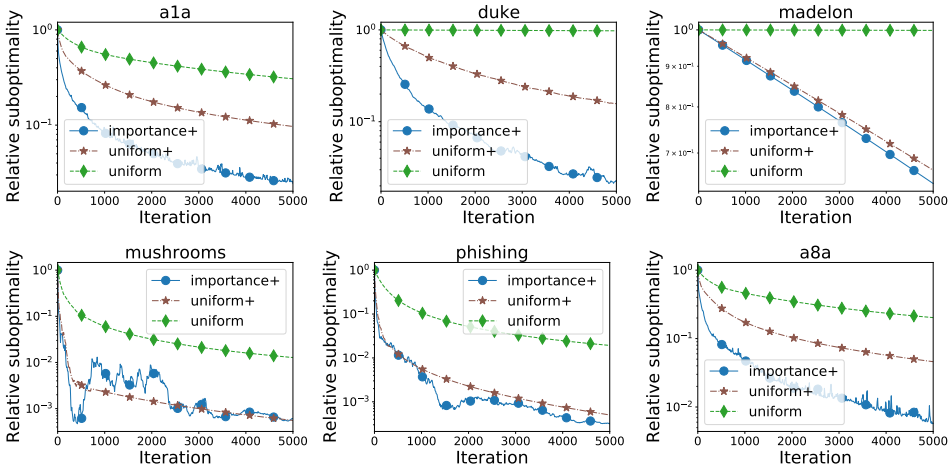

Figure 2: Comparison of our sparsification strategy of size $\tau = 1$ for DIANA+ (Algorithm 2) using i) importance sampling with probabilities (26), ii) uniform sampling with $p_i = (\frac{1}{d}, \frac{1}{d}, \ldots \frac{1}{d})^\top$ and iii) DIANA (Mishchenko et al., 2019) using standard sparsification scheme with uniform sampling. All methods are run with stepsizes as dictated by theory.

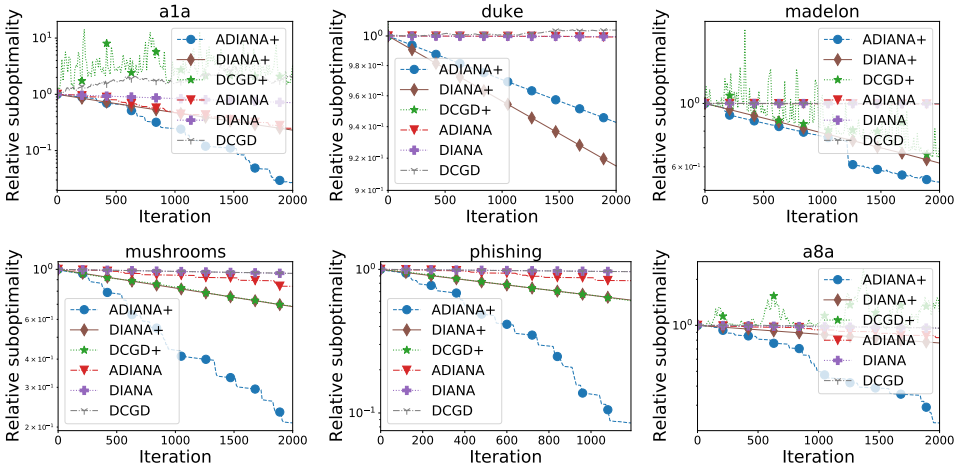

Figure 3: Comparison of the three original methods DCGD (Khirirat et al., 2018), DIANA (Mishchenko et al., 2019) and ADIANA (Li et al., 2020) with the proposed new methods DCGD+ (Alg. 1), DIANA+ (Alg. 2) and ADIANA+ (Alg. 3). All six methods use uniform sampling with single mini-batch size $\tau = 1$.

smaller than some constant. The value of such constant depends on various factors such as the type of sampling and the specific smoothness structure of the objective.

To contrast this with known results, Mishchenko et al. (2020) show that the sparsification does not hurt ISEGA significantly (a method with sparsification unaware of smoothness matrix) as soon as $\tau n \geq d$. Addmitedly, Mishchenko et al. (2020) assume identical smoothness constants for both f and f_i, so such a conclusion is slightly imprecise. In our case, ignoring the $\tilde{\omega}_{\max}$ factor, the rate is dominated by the sparsification factors only if $L = \mathcal{O}\left(\frac{\tilde{\mathcal{L}}_{\max}}{n}\right)$.

The results are presented in Fugure 4 (Iteration vs Residual) and Fugure 5 (Communication vs Residual). As expected, we see that the sparsification only hurts the iteration complexity when τ is below certain treshold which is smaller for the uniform sampling compared to the importance sampling. Consequently, DIANA+ is capable of significantly reducing the worker->server communication at no cost in terms of the total iteration complexity.

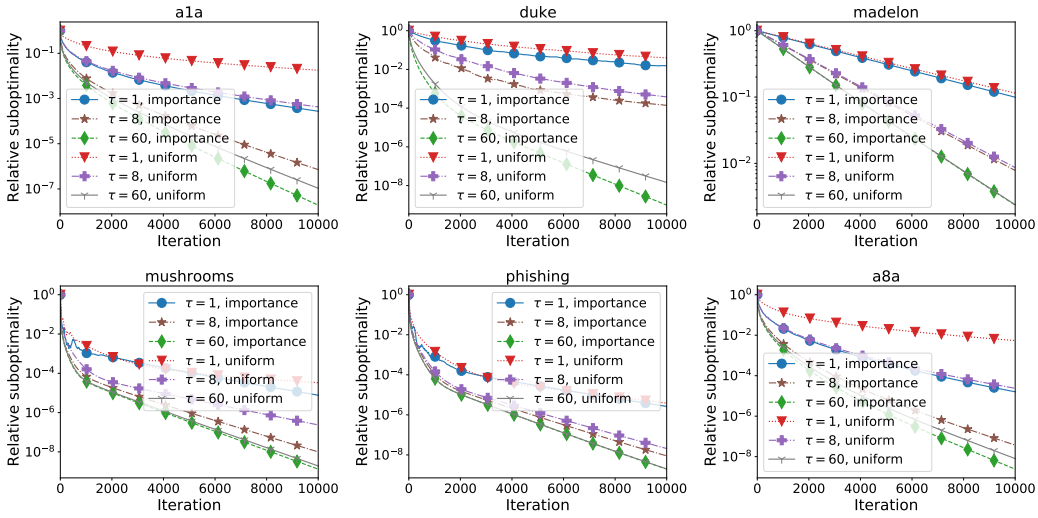

Figure 4: Effect of τ on the convergence speed of DIANA+ (Algorithm 2).

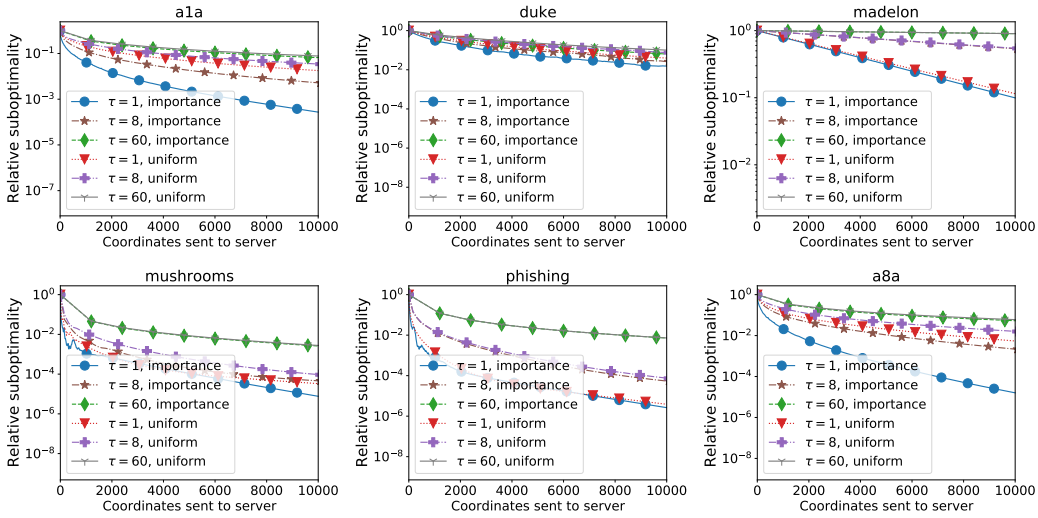

Figure 5: Same as Figure 4, but x-axis corresponds to the coordinates sent to the server instead of the iteration.

G Conclusions, Extensions and Future Work

In this paper we have proposed a novel gradient sparsification technique for distributed optimization and demonstrated that it allows one to properly exploit the smoothness structure of the local objective. We have shown that the proposed matrix-smoothness-aware sparsification can be coupled with both the variance reduction and acceleration, providing further speedup in terms of the convergence rate and the total bits transmitted from workers to server. Next, we list possible extensions of our work that we believe can or should be done in the future:

- **Subsampling the local objective.** While DCGD+, DIANA+ and ADIANA+ all require an access to the full local gradient from each machine at every iteration, we believe this requirement can be easily dropped. In particular, the local objective can be further subsampled and extra variance reduction can be employed on top of these methods, similarly to as done for ISAEGA (Hanzely & Richtárik, 2019b).

- **Greedy sparsification.** Notice that the sparsified local gradient can be seen as a randomized coordinate descent estimator of a given machine. However, greedy coordinate descent was shown to outperform randomized coordinate descent in certain scenarios (Nutini et al., 2017). Therefore, one might pose a question whether a greedy sparsification might work for distributed optimization.
- **Bi-directional sparsification.** As we also mention in Section H, one drawback of our approach[8] is that only worker→server communication is sparse. It would be very interesting to develop a bi-directional sparsification capable of properly exploiting the smoothness matrices. For this matter, in Section O we develop and analyze DIANA++ method employing bi-directional matrix-smoothness-aware sparsification and twofold variance reduction.
- **Weakly convex and non-convex cases.** While we state our theory for the strongly convex case (i.e., Assumpiton 3), it can be rather easily extended to weakly convex case (i.e., $\mu = 0$). However, obtaining an efficiennt smoothness matrix aware sparsification for non-convex optimization remains an open problem.

H LIMITATIONS

Next, we discuss main limitations of our approach.

- The server is required to store matrices $\mathbf{L}_i^{1/2}$ for all machines $i \in [n]$ and multiply them by sparse updates $\mathbf{C}_i^k \mathbf{L}_i^{\dagger 1/2} \nabla f_i(x^k)$ in each iteration. Therefore, our method is not expected to be practical when d is large and matrices $\mathbf{L}_i$ are not of a special structure so that they are cheap to store and so that $\mathbf{C}_i^k \mathbf{L}_i^{\dagger 1/2} \nabla f_i(x^k)$ can be evaluated cheaply.[9]. On the other hand, our strategy is still practical when i) d is small or ii) $\mathbf{L}_i$ is of a special structure such as low rank or diagonal. In particular, diagonal $\mathbf{L}_i$ requires only $\mathcal{O}(\tau)$ extra computation per each node (which is negligible), while attaining a rate which is never worse compared to the naive sparsification.
- Except DIANA++ method presented in Section O, we sparsify only the communication from the workers to server. Sparsifying workers→server communication only is very common in the area of distributed optimization as the workers→server communication is significantly more expensive compared to the server→workers communication. Such a phenomenon can be assigned to the fact that the server is broadcasting the same vector to all workers, and thus the server→workers communication can be implemented more efficiently.

Remark 6. *The overhead that comes from the computation of $\mathbf{L}_i^{\dagger 1/2} \nabla f_i(x^k)$ is not an issue in general. Given that $\mathbf{L}_i$ is of rank r, one requires $\mathcal{O}\left(d^2 r\right)$ flops to precompute SVD of $\mathbf{L}_i$. Given that SVD of $\mathbf{L}_i$ is known, the evaluation of $\mathbf{L}_i^{\dagger 1/2} \nabla f_i(x^k)$ takes only $\mathcal{O}\left(d^2 r\right)$ flops. While the cost of computing $\nabla f_i(x^k)$ varies depending on the application, we can expect it to takes at least $\Omega\left(d^2 r\right)$ flops for the application of generalized linear models (i.e., logistic regression). Next, we shall mention that evaluating $\mathbf{L}_i^{\dagger 1/2} \nabla f_i(x^k)$ comes at $\mathcal{O}\left(d\right)$ cost when $\mathbf{L}_i$ is diagonal.*

[8] In fact, this is a drawback of the vast majority of compression methods from the literature. A notable exception is DoubleSqueeze (Tang et al., 2019) which compresses the server→worker communication too.

[9] For example, if $\mathbf{L}_i$ is of rank r, for all i, we require extra $\mathcal{O}(ndr)$ storage and $\mathcal{O}(ndr)$ flops at the server at each iteration.

I TABLE OF FREQUENTLY USED NOTATION

Table 5: Notation used throughout the paper

Symbol	Description	Reference
d	dimension of the model $x \in \mathbb{R}^d$	(29)
μ	strong convexity parameter of f	Asm. 3
$\mathbf{L}$	smoothness matrix of f	Asm. 2
$\mathbf{L}_{ij}$	the element at ith row and jth column of $\mathbf{L}$	-
$\mathbf{L}_i$	smoothness matrix of f_i	Asm. 2
L_i	smoothness constant of $f_i(x)$, i.e., $L_i = \lambda_{\max}(\mathbf{L}_i)$	-
L	smoothness constant of f, i.e., $L = \lambda_{\max}(\mathbf{L})$	-
S	random sampling (subset) of coordinates $[d] \coloneqq \{1, 2, \ldots, d\}$	-
$p_{jl},\ p_j$	$p_{jl} \coloneqq \mathrm{Prob}\left(\{j, l\} \subseteq S\right),\ p_j \coloneqq p_{jj}$	-
$\mathbf{P}$	the probability matrix $(p_{jl})_{j,l=1}^d$ associated with random sampling S	(15)
v_i	ESO parameters associated with f and S jointly	-
$\mathbf{C}$	diagonal sketch matrix with ith random variable $\mathrm{c}_i = {}^1/_{p_i}$ if $i \in S$ and 0 otherwise	(13)
ω	variance of general compression operator $\mathcal{C}$, i.e. $\mathbb{E}\left[\|\mathcal{C}(x) - x\|^2\right] \leq \omega\|x\|^2,\ \forall x \in \mathbb{R}^d$	-
$\overline{\mathbf{C}},\ \overline{\mathbf{C}}_i^k$	$\overline{\mathbf{C}} \coloneqq \mathbf{L}^{1/2}\mathbf{C}\mathbf{L}^{\dagger 1/2},\ \overline{\mathbf{C}}_i^k = \mathbf{L}_i^{1/2}\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}$	-
$\mathbf{I},\ \mathbf{E}$	the identity matrix and the matrix with all entries equal to 1	-
$\overline{\mathbf{P}},\ \widetilde{\mathbf{P}}$	$\overline{\mathbf{P}} = \mathbf{Diag}\left({}^1/_p\right)\mathbf{P}\mathbf{Diag}\left({}^1/_p\right)$ with entries $\bar{p}_{ij} = \frac{p_{ij}}{p_i p_j}$, and $\widetilde{\mathbf{P}} = \overline{\mathbf{P}} - \mathbf{E}$	(15)
$\overline{\mathcal{L}},\ \widetilde{\mathcal{L}}$	expected smoothness constants $\overline{\mathcal{L}} = \lambda_{\max}(\overline{\mathbf{P}} \circ \mathbf{L}),\ \widetilde{\mathcal{L}} = \lambda_{\max}(\widetilde{\mathbf{P}} \circ \mathbf{L})$	-
n	number of parallel machines in distributed setting	(8)
$\mathbf{C}_i, \mathbf{P}_i, \overline{\mathbf{P}}_i, \widetilde{\mathbf{P}}_i$	diagonal sketch matrix and probability matrices for ith worker	(13), (15)
$p_{i;j},\ \bar{p}_{i;j},\ \widetilde{p}_{i;j}$	j-th diagonal element of $\mathbf{P}_i,\ \overline{\mathbf{P}}_i,\ \widetilde{\mathbf{P}}_i$	-
ω_i	variance of compression operator induced by $\mathbf{C}_i$, i.e. $\omega_i = \max_{1 \leq j \leq d} \frac{1}{p_{i;j}} - 1$	-
$\omega_{\max}$	$\max_{1 \leq i \leq n} \omega_i = \max_{1 \leq i \leq n} \max_{1 \leq j \leq d} \frac{1}{p_{i;j}} - 1$	(19)
$\overline{\mathcal{L}}_i,\ \widetilde{\mathcal{L}}_i$	expected smoothness constants, $\overline{\mathcal{L}}_i = \lambda_{\max}(\overline{\mathbf{P}}_i \circ \mathbf{L}_i),\ \widetilde{\mathcal{L}}_i = \lambda_{\max}(\widetilde{\mathbf{P}}_i \circ \mathbf{L}_i)$	-
$\overline{\mathcal{L}}_{\max},\ \widetilde{\mathcal{L}}_{\max}$	$\overline{\mathcal{L}}_{\max} = \max_{1 \leq i \leq n} \lambda_{\max}(\overline{\mathbf{P}}_i \circ \mathbf{L}_i),\ \widetilde{\mathcal{L}}_{\max} = \max_{1 \leq i \leq n} \lambda_{\max}(\widetilde{\mathbf{P}}_i \circ \mathbf{L}_i)$	(16)
$\nu,\ \nu_s$	Parameters describing distribution of $\mathbf{L}_i$, $\nu \coloneqq \frac{\sum_{i=1}^n L_i}{\max_{i \in [n]} L_i}, \nu_s \coloneqq \max_{i \in [n]} \frac{\sum_{j=1}^d \mathbf{L}_{i;j}^{1/s}}{\max_{j \in [d]} \mathbf{L}_{i;j}^{1/s}}$,	(21)

J THEORY IN THE SINGLE NODE CASE: RCD AS SKETCHED GRADIENT DESCENT (SKGD)

In single node setup, matrix smoothness assumption and arbitrary samplings have been considered mainly in the context of coordinate descent methods. For example, randomized sampling $S = \{j\}, j \in [d]$ with arbitrary probabilities $p_j \in (0, 1]$ reduces to standard *Randomized Coordinate Descent (RCD)* algorithms (Nesterov, 2012; Richtárik & Takáč, 2014). Parallel and mini-batch variants arise when the sampling S contains more than one coordinate (Bradley et al., 2011; Richtárik & Takáč, 2016b). The first coordinate descent method analyzed with arbitrary sampling and under $\mathbf{L}$-smoothness assumption is the 'NSync algorithm (Richtárik & Takáč, 2016a; Qu & Richtárik, 2016a;b) considered for strongly convex losses. In the same general setup, Hanzely & Richtárik (2019a) developed and analyzed *Accelerated Coordinate Descent*. Recently, Hanzely et al. (2018) developed a variance reduced coordinate descent algorithm, *SEGA (SkEtched GrAdient)*, which uses general sketch matrices and handles non-separable proximal terms in contrast to traditional coordinate descent methods. This idea of gradient sketching then extended to *Generalized Jacobian Sketching (GJS)* algorithm providing a unified theory for first-order methods with variance reduced (Hanzely & Richtárik, 2019b).

Consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^d} f(x), \tag{29}$$

with very large dimension d and assume that function f is $\mathbf{L}$-smooth. In this setting, the state-of-art methods are *Randomized Coordinate Descent (RCD)* type methods where in each iteration only a few coordinates get updated. Here we present new theories for RCD with arbitrary sampling paradigm, which are new and follow the idea of sketches. We will view RCD as a special case of *Compressed Gradient Descent (CGD)* with sketches (13).

J.1 'NSYNC

First, we recall the first coordinate descent type algorithm, 'NSync (Richtárik & Takáč, 2016a), using arbitrary sampling. Let $S \subseteq [d]$ be an arbitrary (proper) sampling[10] of coordinates such that $p_j \coloneqq \text{Prob}(j \in S) > 0$, $j = 1, 2, \ldots, d$. For a vector $h \in \mathbb{R}^d$, let $h_S \in \mathbb{R}^d$ be the vector coinciding with h at coordinates $j \in S$ and zeros everywhere else. Denote by $\circ$ the Hadamard (i.e. element-wise) product. Given an arbitrary sampling S and smoothness matrix $\mathbf{L}$, let $v = (v_1, v_2, \ldots, v_d)$ be positive constants satisfying the *Expected Separable Overapproximation (ESO)* inequality

$$\mathbf{P} \circ \mathbf{L} \preceq \mathbf{Diag}(p \circ v), \tag{30}$$

where $\mathbf{P}$ is the probability matrix associated with sampling S having entries $p_{jl} \coloneqq \text{Prob}(\{j, l\} \subseteq S)$, $p_j = p_{jj}$. Analogous to (15), let $\widetilde{\mathbf{P}} = \overline{\mathbf{P}} - \mathbf{E}$.

Algorithm 4 'NSYNC (RICHTÁRIK & TAKÁČ, 2016A)

1: **Input:** Initial point $x^0 \in \mathbb{R}^d$, random sampling S, step size parameters v, current point x^k
2: Sample random set of coordinates $S_k \sim S$
3: Update selected coordinates $x^{k+1} = x^k - \frac{1}{v} \circ \nabla f(x)_{S_k}$

Theorem 8 ('NSync, (Richtárik & Takáč, 2016a)). *Let Assumptions 2, 3 hold and $v \sim ESO(f, S)$ be the vector of ESO parameters associated with function f and sampling S. Then the iterates $\{x^k\}$ of 'NSync converge as follows*

$$\mathbb{E}\left[f(x^k)\right] - f(x^*) \leq \left(1 - \min_{1 \leq j \leq d} \frac{p_j \mu}{v_j}\right)^k \Delta_f,$$

where $\Delta_f = f(x^0) - f(x^*)$.

[10] only proper samplings are considered in this work

Thus, 'Nsync gives an iteration complexity

$$\max_{1\leq j\leq d} \frac{v_j}{p_j\mu} \log \frac{\Delta_f}{\varepsilon}. \tag{31}$$

In case of serial sampling, namely $|S| = 1$ a.s., we have $\mathbf{P} = \mathbf{Diag}(p_1, p_2, \ldots, p_d)$. Hence ESO holds with $v_j = \mathbf{L}_{jj}$ and iteration complexity becomes $\max_j \frac{\mathbf{L}_{jj}}{p_j\mu} \log \frac{\Delta_f}{\varepsilon}$. This leads to the optimal probabilities $p_j = \frac{\mathbf{L}_{jj}}{\sum_l \mathbf{L}_{ll}}$ yielding iteration complexity $\frac{\sum \mathbf{L}_{jj}}{\mu} \log \frac{\Delta_f}{\varepsilon}$.

J.2 Sketched Gradient Descent (SkGD)

Let us view RCD methods as a special case of *Compressed Gradient Descent (CGD)* with linear and diagonal sketch $\mathbf{C}$ defined in (13) and consider random sparsification operator $\mathcal{C}$ induced by random diagonal sketch $\mathbf{C}$, namely $\mathcal{C}(x) = \mathbf{C}x$, $x \in \mathbb{R}^d$. Clearly, $\mathcal{C}$ is an unbiased compression (i.e. $\mathbb{E}[\mathcal{C}(x)] = x$) with variance $\omega = \max_{1\leq j\leq d} \frac{1}{p_j} - 1$:

$$\mathbb{E}\left[\|\mathbf{C}x - x\|_2^2\right] = x^\top \mathbb{E}\left[\mathbf{C}^2 - \mathbf{I}\right] x \leq \omega\|x\|_2^2. \tag{32}$$

Algorithm 5 SkGD

1: **Input:** Initial point $x^0 \in \mathbb{R}^d$, diagonal sketch $\mathbf{C}$, step size γ, current point x^k
2: $x^{k+1} = x^k - \gamma\mathbf{C}\nabla f(x^k)$

Theorem 9 (see L.1)**.** *Let Assumptions 2, 3 hold and S be any proper sampling with probability matrix $\mathbf{P}$. Then, for the step-size $0 < \gamma \leq \lambda_{\max}^{-1}\left(\overline{\mathbf{P}} \circ \mathbf{L}\right)$, the iterates $\{x^k\}$ of Algorithm 5 converge as follows*

$$\mathbb{E}\left[f(x^k)\right] - f(x^*) \leq (1 - \gamma\mu)^k \Delta_f.$$

The following lemma shows that, both 'NSync and SkGD provide the same theoretical guarantees.

Lemma 10.

$$\min_{v:\ \mathbf{P}\circ\mathbf{L}\leq\mathbf{Diag}(v\circ p)} \max_{1\leq j\leq d} \frac{v_j}{p_j} = \lambda_{\max}\left(\overline{\mathbf{P}} \circ \mathbf{L}\right).$$

Proof. If parameters v satisfy ESO inequality (30), then parameters defined by

$$v_i' := p_i \max_j \frac{v_j}{p_j} \geq v_i, \quad 1 \leq i \leq d$$

also satisfy ESO inequality and give the same iteration complexity as

$$\lambda := \max_i \frac{v_i}{p_i} = \max_i \frac{v_i'}{p_i}.$$

In particular, this implies that instead of searching for d parameters $v_1, \ldots, v_d$ satisfying ESO inequality $\mathbf{P} \circ \mathbf{L} \leq \mathbf{Diag}(v \circ p)$ it suffices to find one scalar $\lambda > 0$ such that $\mathbf{P} \circ \mathbf{L} \leq \mathbf{Diag}(\lambda p \circ p)$ and set $v_i = \lambda p_i$ for all $i \in [d]$. The optimal (smallest) value of the scaling factor is

$$\lambda = \lambda_{\max}\left(\mathbf{Diag}(1/p)(\mathbf{P} \circ \mathbf{L})\mathbf{Diag}(1/p)\right) = \lambda_{\max}\left((\mathbf{Diag}(1/p)\mathbf{P}\mathbf{Diag}(1/p)) \circ \mathbf{L}\right) = \lambda_{\max}\left(\overline{\mathbf{P}} \circ \mathbf{L}\right).$$

Notice that with the choice of $v = \lambda p$, iteration complexities as well as the update rules of both methods coincide. □

One difference between these two methods is that, the update direction $\frac{1}{v} \circ \nabla f(x)_S$ of 'NSync is biased in general as opposed to unbiased direction $\frac{1}{p} \circ \nabla f(x)_S$ of SkGD.

Note that the rate and the analysis of Theorem 9 is with respect to functional values (i.e. $f(x^k) - f^*$). Natural question is to develop an analysis based on iterates of the algorithm (i.e. $\|x^k - x^*\|^2$). Below, we provide such analysis under slightly different conditions on f and with weighted distances. Formally, let, instead of $\mathbf{L}$-smoothness and μ-convexity, assume

$$\mu\|x - x^*\|_{\mathbf{L}}^2 + \|\nabla f(x)\|^2 \leq 2\langle\nabla f(x), (x - x^*)\rangle_{\mathbf{L}}. \tag{33}$$

Notice that the following is true just by combining $\mathbf{L}$-smoothness and μ-convexity:

$$\mu\|x-x^*\|^2 + \|\nabla f(x)\|^2_{\mathbf{L}^\dagger} \leq 2\langle \nabla f(x), (x-x^*)\rangle. \tag{34}$$

However, in general, inequalities (33) and (34) are not equivalent.

Theorem 11. *Let instead of $\mathbf{L}$-smoothness and μ-convexity assume (33) holds. Then, for the step-size $0 < \gamma \leq \lambda_{\max}^{-1}\left(\overline{\mathbf{P}} \circ \mathbf{L}\right)$, the iterates $\{x^k\}$ of Algorithm 5 converge as follows*

$$\mathbb{E}\left[\|x^k - x^*\|^2_{\mathbf{L}}\right] \leq (1-\gamma\mu)^k \|x^0 - x^*\|^2_{\mathbf{L}}.$$

Proof. Consider the improvement of the algorithm in a single iteration $x^+ = x - \gamma \mathbf{C}\nabla f(x)$.

$$\begin{aligned}
\mathbb{E}\left[\|x^+ - x^*\|^2_{\mathbf{L}}\right] &= \mathbb{E}\left[\|x - x^* - \gamma\mathbf{C}\nabla f(x)\|^2_{\mathbf{L}}\right] \\
&= \|x-x^*\|^2_{\mathbf{L}} - 2\gamma\langle x-x^*, \nabla f(x)\rangle_{\mathbf{L}} + \gamma^2 \mathbb{E}\left[\|\mathbf{C}\nabla f(x)\|^2_{\mathbf{L}}\right] \\
&= \|x-x^*\|^2_{\mathbf{L}} - 2\gamma\langle x-x^*, \nabla f(x)\rangle_{\mathbf{L}} + \gamma^2 \|\nabla f(x)\|^2_{\mathbb{E}[\mathbf{CLC}]} \\
&\overset{(45)}{=} \|x-x^*\|^2_{\mathbf{L}} - 2\gamma\langle x-x^*, \nabla f(x)\rangle_{\mathbf{L}} + \gamma^2 \|\nabla f(x)\|^2_{\overline{\mathbf{P}}\circ\mathbf{L}} \\
&\leq \|x-x^*\|^2_{\mathbf{L}} - 2\gamma\langle x-x^*, \nabla f(x)\rangle_{\mathbf{L}} + \gamma^2 \lambda_{\max}(\overline{\mathbf{P}}\circ\mathbf{L})\|\nabla f(x)\|^2 \\
&\leq \|x-x^*\|^2_{\mathbf{L}} - 2\gamma\langle x-x^*, \nabla f(x)\rangle_{\mathbf{L}} + \gamma \|\nabla f(x)\|^2 \\
&\overset{(33)}{\leq} (1-\gamma\mu)\|x-x^*\|^2_{\mathbf{L}}.
\end{aligned}$$

□

J.3 CGD+

Here we introduce a new variant of CGD with non-diagonal matrix $\overline{\mathbf{C}} := \mathbf{L}^{1/2}\mathbf{C}\mathbf{L}^{\dagger 1/2}$, which works with any proximable regularizer $R(x)$. In this case the method converges to the neighborhood of the solution. Recall that the proximal operator is defined as followsL:

$$\mathrm{prox}_R(x) = \arg\min_{u\in\mathbb{R}^d}\left(R(u) + \frac{1}{2}\|u-x\|^2\right). \tag{35}$$

Define expected smoothness constants

$$\overline{\mathcal{L}} = \lambda_{\max}(\overline{\mathbf{P}}\circ\mathbf{L}), \quad \widetilde{\mathcal{L}} = \lambda_{\max}(\widetilde{\mathbf{P}}\circ\mathbf{L}).$$

The following lemma reveals the relationship between these constants.

Lemma 12. *Let $L = \lambda_{\max}(\mathbf{L})$. Then $L \leq \overline{\mathcal{L}} \leq L + \widetilde{\mathcal{L}}$.*

Proof. First, positive semi-definiteness of $\mathbf{P}$ was proved in Theorem 3.1 (Qu & Richtárik, 2016). As $\mathbf{Diag}(1/p)$ is positive definite, then $\overline{\mathbf{P}}$ is positive semi-definite too. Since Hadamard product $\circ$ preserves positive semi-definiteness, we have that $\overline{\mathbf{P}}\circ\mathbf{L} \succeq 0$. It follows from Lemma 18 that

$$\mathbb{E}\left[\mathbf{L}^{1/2}\left(\overline{\mathbf{C}}-\mathbf{I}\right)^\top\left(\overline{\mathbf{C}}-\mathbf{I}\right)\mathbf{L}^{1/2}\right] = \mathbf{L}^{1/2}\mathbf{L}^{\dagger 1/2}(\widetilde{\mathbf{P}}\circ\mathbf{L})\mathbf{L}^{\dagger 1/2}\mathbf{L}^{1/2}.$$

Hence the left hand side as well as $\widetilde{\mathbf{P}}\circ\mathbf{L}$ are symmetric and positive semidefinite. In particular, $\overline{\mathbf{P}}\circ\mathbf{L} \succeq \mathbf{L}$. Hence $L = \lambda_{\max}(\mathbf{L}) \leq \lambda_{\max}(\overline{\mathbf{P}}\circ\mathbf{L}) = \overline{\mathcal{L}}$. The upper bound follows from the convexity of $\lambda_{\max}$ as $\overline{\mathcal{L}} = \lambda_{\max}(\overline{\mathbf{P}}\circ\mathbf{L}) = \lambda_{\max}(\mathbf{L} + \widetilde{\mathbf{P}}\circ\mathbf{L}) \leq \lambda_{\max}(\mathbf{L}) + \lambda_{\max}(\widetilde{\mathbf{P}}\circ\mathbf{L}) = L + \widetilde{\mathcal{L}}$. □

Algorithm 6 CGD+

1: **Input:** Initial point $x^0 \in \mathbb{R}^d$, sketch matrix $\overline{\mathbf{C}} = \mathbf{L}^{1/2}\mathbf{C}\mathbf{L}^{\dagger 1/2}$, step size γ, current point x^k
2: $x^{k+1} = \mathrm{prox}_{\gamma R}\left(x^k - \gamma\overline{\mathbf{C}}\nabla f(x^k)\right)$

With the new sketch $\overline{\mathbf{C}}$ in Algorithm 6 we able to perform the analysis with respect to iterates in standard norm, under strong convexity and $\mathbf{L}$-smoothness, allowing any proximable regularizer.

Table 6: Original and proposed new methods for both single node and distributed setups.

ORIGINAL	'NSYNC	CGD	DCGD	DIANA	ADIANA
NEW	SKGD (ALG.5)	CGD+ (ALG.6)	DCGD+ (ALG.1)	DIANA+ (ALG.2)	ADIANA+ (ALG.3)
PROXIMAL	✗	✓	✓	✓	✓
DISTRIBUTED	✗	✗	✓	✓	✓
VARIANCE REDUCED	✗	✗	✗	✓	✓
ACCELERATED	✗	✗	✗	✗	✓

Table 7: Complexity of new methods with hidden log factors and constants.

Method	Iteration Complexity
SkGD (Algorithm 5)	$\frac{\overline{\mathcal{L}}}{\mu}$
CGD+ (Algorithm 6)	$\frac{\overline{\mathcal{L}}}{\mu} + \frac{\widetilde{\mathcal{L}}}{\mu^2 \varepsilon}$
DCGD+ (Algorithm 1)	$\frac{L}{\mu} + \frac{\widetilde{\mathcal{L}}_{\max}}{\mu n} + \frac{\widetilde{\mathcal{L}}_{\max}}{\mu^2 n \varepsilon}$
DIANA+ (Algorithm 2)	$\omega_{\max} + \frac{L}{\mu} + \frac{\widetilde{\mathcal{L}}_{\max}}{\mu n}$
ADIANA+ (Algorithm 3)	$\begin{cases} \omega_{\max} + \sqrt{\omega_{\max} \frac{\widetilde{\mathcal{L}}_{\max}}{\mu n}} & \text{if} \quad nL \leq \widetilde{\mathcal{L}}_{\max} \\ \omega_{\max} + \sqrt{\frac{L}{\mu}} + \sqrt{\omega_{\max} \sqrt{\frac{\widetilde{\mathcal{L}}_{\max}}{\mu n}} \sqrt{\frac{L}{\mu}}} & \text{if} \quad nL > \widetilde{\mathcal{L}}_{\max}. \end{cases}$

Theorem 13 (see L.2)**.** *Let Assumptions 2, 3 hold and S be a sampling with probability matrix $\mathbf{P}$. Then, for the step-size $0 < \gamma \leq 1/2\overline{\mathcal{L}}$, the iterates $\{x^k\}$ of Algorithm 6 converge as follows*

$$\mathbb{E}\left[\|x^k - x^*\|^2\right] \leq (1 - \gamma\mu)^k \|x^0 - x^*\|^2 + \frac{2\gamma\widetilde{\mathcal{L}}}{\mu} \|\nabla f(x^*)\|^2_{\mathbf{L}^\dagger}.$$

K LOWER BOUNDS FOR SKETCHES AS LINEAR COMPRESSION OPERATORS

Here we investigate general sketch matrices $\mathbf{S}$ as a linear compression operators. The motivation of this is to understand the trade-off between communication and variance of linear compressors. The notation, used in this section only, slightly deviates from the paper but otherwise is consistent throughout the section.

Consider compression of vectors $x \in \mathbb{R}^d$ allowing approximation error in exchange for less bits of communication. Let compression operator $\mathcal{C}\colon \mathbb{R}^d \to \mathbb{R}^d$ be composed of some linear encoder $E(x) = \mathbf{S}x$ with $s \times d$ sketch matrix $\mathbf{S}$ and an arbitrary decoder $D\colon \mathbb{R}^s \to \mathbb{R}^d$, so that $\mathcal{C}(x) = D(\mathbf{S}x)$. Throughout we consider the space $\mathbb{R}^d$ equipped with an inner product together with its induced norm given by some symmetric and positive definite matrix $\mathbf{B}$ of size $d \times d$ as follows

$$\langle x, y\rangle_{\mathbf{B}} = x^\top \mathbf{B} y, \quad \|x\|_{\mathbf{B}} = \sqrt{\langle x, x\rangle_{\mathbf{B}}}, \quad x, y \in \mathbb{R}^d.$$

In general, we let matrix $\mathbf{S}$, number of rows s and decoder D to be random, while the matrix $\mathbf{B}$ will be fixed throughout the analysis. Since we consider only linear encoders, we may assume $\|x\|_{\mathbf{B}} = 1$.

K.1 FIXED SKETCHES

We first analyze the case where the sketch matrix $\mathbf{S}$ is fixed and hence the compression operator $\mathcal{C}$ is deterministic. The analysis then we will lead us on a more usefull result for random sketches. The decoder D receiving vector $y = \mathbf{S}x$ should be able to reconstruct $\hat{x} = D(y)$ so to minimize the squared error

$$\alpha(\mathbf{S}) \coloneqq \sup_{\|x\|_{\mathbf{B}}=1} \|\mathcal{C}(x) - x\|_{\mathbf{B}}^2 = \sup_{\|x\|_{\mathbf{B}}=1} \|D(\mathbf{S}x) - x\|_{\mathbf{B}}^2 \leq 1.$$

The following lemma shows the optimal strategy for the decoder and possible values for $\alpha(\mathbf{S})$.

Lemma 14. *For a fixed sketch $\mathbf{S}$ the optimal reconstruction from $y = \mathbf{S}x$ is*

$$D^*(y) = \mathbf{S}^{\dagger_{\mathbf{B}}} y \equiv \mathbf{B}^{-1}\mathbf{S}^\top \left(\mathbf{S}\mathbf{B}^{-1}\mathbf{S}^\top\right)^\dagger y, \tag{36}$$

where $\cdot^\dagger$ indicates the Moore–Penrose inverse of a matrix. Furthermore, if $\ker(\mathbf{S}) = \{0\}$ then $\alpha(\mathbf{S}) = 0$ as in this case $D^*(\mathbf{S}x) = x$ for any $x \in \mathbb{R}^d$. Otherwise, if $\ker(\mathbf{S}) \neq \{0\}$, then $\alpha(\mathbf{S}) = 1$.

Proof. Let $\ker(\mathbf{S}) = \{z\colon \mathbf{S}z = 0\}$ be the kernel of $\mathbf{S}$ and $x^{\dagger_{\mathbf{B}}} = \mathbf{S}^{\dagger_{\mathbf{B}}} y$ be the minimal $\mathbf{B}$-norm solution to the system $\mathbf{S}z = y$ so that the set of all solutions is $x^{\dagger_{\mathbf{B}}} + \ker(\mathbf{S})$:

$$x^{\dagger_{\mathbf{B}}} = \operatorname*{arg\,min}_{x\colon \mathbf{S}x = y} \|x\|_{\mathbf{B}}^2 = \mathbf{S}^{\dagger_{\mathbf{B}}} y = \mathbf{B}^{-1/2}\left(\mathbf{S}\mathbf{B}^{-1/2}\right)^\dagger y,$$

Denote by

$$\hat{S}(x) \coloneqq \left(x^{\dagger_{\mathbf{B}}} + \ker(\mathbf{S})\right) \cap \{z \in \mathbb{R}^d\colon \|z\|_{\mathbf{B}} = 1\}$$

the intersection of the affine set of solutions and the unit sphere. Notice that initial vector $x \in \hat{S}(x)$ as it has unit $\mathbf{B}$-norm and satisfies $\mathbf{S}x = y$. Now the cost of sending $\mathbf{S}x$ instead of original x, is the uncertainty that the decoder has to deal with by estimating the original vector within the set $\hat{S}$ so to minimize α. We first show that $x^S \coloneqq 2x^{\dagger_{\mathbf{B}}} - x \in \hat{S}(x)$, which is equivalent to

$$x^{\dagger_{\mathbf{B}}} - x \in \ker(\mathbf{S}) \quad \text{and} \quad \|2x^{\dagger_{\mathbf{B}}} - x\|_{\mathbf{B}}^2 = 1.$$

The first claim follows from the fact that both x and $x^{\dagger_{\mathbf{B}}}$ are solutions to $\mathbf{S}z = y$, namely $\mathbf{S}x^{\dagger_{\mathbf{B}}} = y = \mathbf{S}x$. Expanding the square in the second claim we get $\left\langle x^{\dagger_{\mathbf{B}}}, x^{\dagger_{\mathbf{B}}} - x\right\rangle_{\mathbf{B}} = 0$ which holds as $x^{\dagger_{\mathbf{B}}}$ is the minimal $\mathbf{B}$-norm solution. Therefore the vector y the decoder receives does not differentiate between x and x^S. This implies that for any choice of $\hat{x}$ of the decoder

$$\max\left(\|\hat{x} - x\|_{\mathbf{B}}^2, \|\hat{x} - x^S\|_{\mathbf{B}}^2\right) \geq \tfrac{1}{4}\left(\|\hat{x} - x\|_{\mathbf{B}} + \|\hat{x} - x^S\|_{\mathbf{B}}\right)^2 \geq \tfrac{1}{4}\|x^S - x\|_{\mathbf{B}}^2 = \|x^{\dagger_{\mathbf{B}}} - x\|_{\mathbf{B}}^2$$

squared-error is unavoidable for the couple x, x^S and the optimal choice is $\hat{x} = x^{\dagger_{\mathbf{B}}}$. Thus, the optimal decoding strategy to $y = \mathbf{S}x$ is $D^*(y) = x^{\dagger_{\mathbf{B}}}$ given in (36). Now, if $\ker(\mathbf{S}) \neq \{0\}$ then we could pick the initial vector x from the kernel space, i.e. $x \in \ker(\mathbf{S})$ and $\|x\|_{\mathbf{B}} = 1$. Then we would have $x^{\dagger_{\mathbf{B}}} = 0$ and hence the minimal squared-error $\alpha(\mathbf{S}) = 1$. On the other hand, if $\ker(S) = \{0\}$, then $x^{\dagger_{\mathbf{B}}} = x$ as the system $\mathbf{S}z = y$ has unique solution. □

To conclude for fixed sketches, notice that, x and x^S are in symmetry in this analysis. Indeed, if the initial vector was x^S as opposed to x, then $\mathbf{S}x = \mathbf{S}x^S$, hence $x^{S\dagger_\mathbf{B}} = x^{\dagger_\mathbf{B}}$ and $x^{SS} = x$. Therefore, the analysis of Lemma 14 leads to the following lower bound for any decoder D and initial vector $x \in \mathbb{R}^d$

$$\max_{z=x,x^S} \|\mathcal{C}(z) - z\|_\mathbf{B}^2 \geq \|x^{\dagger_\mathbf{B}} - x\|_\mathbf{B}^2 = 1 - \|x^{\dagger_\mathbf{B}}\|_\mathbf{B}^2 = 1 - \|\mathbf{Z}x\|_\mathbf{B}^2, \tag{37}$$

where we used orthogonality $\left\langle x^{\dagger_\mathbf{B}}, x^{\dagger_\mathbf{B}} - x \right\rangle_\mathbf{B} = 0$ and defined the random matrix $\mathbf{Z} = \mathbf{Z}(\mathbf{S})$ via

$$\mathbf{Z} \coloneqq \mathbf{S}^{\dagger_\mathbf{B}}\mathbf{S} = \mathbf{B}^{-1/2}\left(\mathbf{S}\mathbf{B}^{-1/2}\right)^\dagger \mathbf{S} = \mathbf{B}^{-1}\mathbf{S}^\top\left(\mathbf{S}\mathbf{B}^{-1}\mathbf{S}^\top\right)^\dagger \mathbf{S}.$$

K.2 Random sketches

Now we turn to the general case when sketch matrix $\mathbf{S}$ is random and drawn from some distribution $\mathcal{D}$, to which both encoder and decoder have access. The number of rows s of $\mathbf{S}$ can also be random. In this case, the decoder D upon receiving random vector $y = \mathbf{S}x$ should estimate possibly randomized $\hat{x} = D(y)$ so to minimize the expected square error

$$\alpha(\mathcal{D}) \coloneqq \sup_{\|x\|_\mathbf{B}=1} \mathbb{E}\left[\|\mathcal{C}(x) - x\|_\mathbf{B}^2\right] \leq 1, \tag{38}$$

where $\mathcal{C}(x) = D(\mathbf{S}x)$ is a random mapping with a source of randomness coming from the distribution $\mathcal{D}$ and decoder D. Below we prove a lower bound for $\alpha(\mathcal{D})$.

Theorem 15. *Let $\mathcal{D}$ be some distribution over $s \times d$ matrices $\mathbf{S}$ allowing variable number of rows $s \in [d]$. Then for any (possibly randomized) compression operator $\mathcal{C}(x) = D(\mathbf{S}x)$ with i.i.d. samples $\mathbf{S} \sim \mathcal{D}$ and $x \in \mathbb{R}^d$ the following lower bound holds*

$$\alpha(\mathcal{D}) + \mathbb{E}_\mathcal{D}\left[r/d\right] \geq 1, \tag{39}$$

where $r = \operatorname{rank}(\mathbf{S})$ is the number of independent rows in $\mathbf{S}$.

Proof. Based on the lower bound (37) obtained from the deterministic case, decoder cannot avoid the error $1 - \|\mathbf{Z}x\|_\mathbf{B}^2$ even in the case of knowing what sketch the encoder used. Therefore minimal expected error $1 - \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\|\mathbf{Z}x\|_\mathbf{B}^2$ is unavoidable for any initial x. This leads to the following bound

$$\begin{aligned}
1 - \alpha(\mathcal{D}) \quad &\leq \quad \inf_{\|x\|_\mathbf{B}=1} \mathbb{E}_\mathcal{D}\left[\|\mathbf{Z}x\|_\mathbf{B}^2\right] \\
&= \quad \inf_{\|x\|_\mathbf{B}=1} \mathbb{E}_\mathcal{D}\left[x^\top\mathbf{Z}^\top\mathbf{B}\mathbf{Z}x\right] \\
&\overset{z=\mathbf{B}^{1/2}x}{=} \quad \inf_{\|z\|=1} \mathbb{E}_\mathcal{D}\left[z^\top\mathbf{B}^{-1/2}\mathbf{Z}^\top\mathbf{B}\mathbf{Z}\mathbf{B}^{-1/2}z\right] \\
&= \quad \inf_{\|z\|=1} z^\top\mathbb{E}_\mathcal{D}\left[\mathbf{B}^{-1/2}\mathbf{Z}^\top\mathbf{B}\mathbf{Z}\mathbf{B}^{-1/2}\right]z \\
&= \quad \lambda_{\min}\left(\mathbb{E}_\mathcal{D}\left[\mathbf{B}^{-1/2}\mathbf{Z}^\top\mathbf{B}\mathbf{Z}\mathbf{B}^{-1/2}\right]\right) \\
&= \quad \lambda_{\min}\left(\mathbb{E}_\mathcal{D}\left[\mathbf{B}^{-1}\mathbf{Z}^\top\mathbf{B}\mathbf{Z}\right]\right) \\
&= \quad \lambda_{\min}\left(\mathbb{E}_\mathcal{D}\left[\mathbf{B}^{-1}\mathbf{S}^\top\left(\mathbf{S}\mathbf{B}^{-1}\mathbf{S}^\top\right)^\dagger\mathbf{S}\right]\right) \\
&= \quad \lambda_{\min}\left(\mathbb{E}_\mathcal{D}\left[\mathbf{Z}\right]\right),
\end{aligned}$$

where the expectation is with respect to $\mathbf{S} \sim \mathcal{D}$. Thus, we obtained the following lower bound:

$$\alpha(\mathcal{D}) + \lambda_{\min}\left(\mathbb{E}_\mathcal{D}\left[\mathbf{S}^{\dagger_\mathbf{B}}\mathbf{S}\right]\right) \geq 1. \tag{40}$$

To prove the inequality (39), it is enough to establish the following upper bound for the minimal eigenvalue

$$\lambda_{\min}\left(\mathbb{E}_\mathcal{D}\left[\mathbf{Z}\right]\right) \leq \mathbb{E}_\mathcal{D}\left[r/d\right].$$

We follow the proof of Lemma 4.2 of Gower & Richtárik (2015) to prove this inequality. It can be easily checked that, using the properties of pseudo-inverse, $\mathbf{Z} = \mathbf{S}^{\dagger_\mathbf{B}}\mathbf{S}$ is an idempotent matrix for any $\mathbf{S}$, namely $\mathbf{Z}^2 = \mathbf{Z}$. This implies that all eigenvalues of $\mathbf{Z}$ are either 0 or 1 as they must satisfy

the same relation $\lambda^2 = \lambda$. Trace $\mathrm{tr}(\mathbf{Z})$ of such matrices coincides with the number of non-zero eigenvalues, which also shows the rank:

$$\mathrm{tr}(\mathbf{Z}) = \sum_{i=1}^{d} \lambda_i(\mathbf{Z}) = \#\{i \in [d] \colon \lambda_i(\mathbf{Z}) \neq 0\} = \mathrm{rank}(\mathbf{Z}). \tag{41}$$

From the properties of pseudo-inverse it follows that $\mathrm{rank}(\mathbf{A}^\dagger\mathbf{A}) = \mathrm{rank}(\mathbf{A}^\dagger) = \mathrm{rank}(\mathbf{A})$ for any matrix $\mathbf{A}$. Hence

$$\begin{aligned}\mathrm{rank}(\mathbf{Z}) = \mathrm{rank}(\mathbf{S}^{\dagger_{\mathbf{B}}}\mathbf{S}) &= \mathrm{rank}\left(\mathbf{B}^{-1/2}\left(\mathbf{S}\mathbf{B}^{-1/2}\right)^\dagger \mathbf{S}\right) \\ &= \mathrm{rank}\left(\left(\mathbf{S}\mathbf{B}^{-1/2}\right)^\dagger \mathbf{S}\mathbf{B}^{-1/2}\right) = \mathrm{rank}\left(\mathbf{S}\mathbf{B}^{-1/2}\right) = \mathrm{rank}\left(\mathbf{S}\right) = r.\end{aligned}$$

Combining with (41) we get $\mathrm{tr}(\mathbf{Z}) = r$. The purpose of expressing the rank as a trace is that in contrast to rank, trace and expectation operators are commutative, which basically follows from the linearity of the expectation:

$$\mathrm{tr}\left(\mathbb{E}_{\mathcal{D}}[\mathbf{Z}]\right) = \mathbb{E}_{\mathcal{D}}\left[\mathrm{tr}(\mathbf{Z})\right]. \tag{42}$$

Using (41), (42) and $\mathrm{tr}(\mathbf{Z}) = r$, we conclude

$$\lambda_{\min}\left(\mathbb{E}_{\mathcal{D}}\left[\mathbf{Z}\right]\right) \leq \frac{1}{d}\sum_{i=1}^{d} \lambda_i\left(\mathbb{E}_{\mathcal{D}}\left[\mathbf{Z}\right]\right) = \frac{\mathrm{tr}\left(\mathbb{E}_{\mathcal{D}}\left[\mathbf{Z}\right]\right)}{d} = \frac{\mathbb{E}_{\mathcal{D}}\left[\mathrm{tr}\left(\mathbf{Z}\right)\right]}{d} = \frac{\mathbb{E}_{\mathcal{D}}[r]}{d},$$

which completes the proof. □

K.3 Optimal sketches

With the knowledge of this new lower bound, here we construct a distribution $\mathcal{D}$ of sketches that will achieve equality in (39). Let $\mathbf{B} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ be the eigendecomposition of the symmetric matrix $\mathbf{B}$, where $\mathbf{\Lambda}$ is diagonal with eigenvalues and $\mathbf{Q}$ is orthogonal with eigenvectors as columns. Let $\mathbf{C}$ be the diagonal sketch of size $d \times d$ corresponding to random sparsification with probabilities $p = (p_i)_{i=1}^d$, namely

$$\mathbf{C} = \mathrm{Diag}(c), \quad c_i = \begin{cases} 1 & \text{with prob.} \quad p_i, \\ 0 & \text{with prob.} \quad 1 - p_i. \end{cases}$$

Define a distribution $\mathcal{D} = \mathcal{D}_p$ of sketches as $\mathbf{S} = \mathbf{C}\mathbf{Q}^\top$ and notice that

$$\mathbb{E}_{\mathcal{D}}\left[\mathrm{rank}(\mathbf{S})\right] = \mathbb{E}_{\mathcal{D}}\left[\mathrm{rank}(\mathbf{C})\right] = \mathbb{E}_{\mathcal{D}}\left[\#\{i \in [d] \colon c_i = 1\}\right] = \mathbb{E}_{\mathcal{D}}\left[\sum_{i=1}^{d} c_i\right] = \sum_{i=1}^{d} \mathbb{E}_{\mathcal{D}}\left[c_i\right] = \sum_{i=1}^{d} p_i.$$

Therefore, $\mathbb{E}_{\mathcal{D}}\left[r/d\right] = \frac{1}{d}\sum p_i$. With decoder $D(x) = \mathbf{Q}x$ we get a compression operator $\mathcal{C}(x) = \mathbf{Q}\mathbf{S}x$. Next, we compute $\alpha(\mathcal{D})$ as follows

$$\begin{aligned}
\alpha(\mathcal{D}) &= \sup_{\|x\|_{\mathbf{B}}=1} \mathbb{E}\left[\|\mathcal{C}(x) - x\|_{\mathbf{B}}^2\right] \\
&= \sup_{\|x\|_{\mathbf{B}}=1} \mathbb{E}\left[\|\mathbf{Q}\mathbf{S}x - x\|_{\mathbf{B}}^2\right] \\
&= \sup_{x^\top\mathbf{B}x=1} \mathbb{E}\left[x^\top(\mathbf{I} - \mathbf{Q}\mathbf{S})^\top\mathbf{B}(\mathbf{I} - \mathbf{Q}\mathbf{S})x\right] \\
&= \sup_{x^\top\mathbf{Q}\mathbf{C}\mathbf{Q}^\top x=1} x^\top\mathbb{E}\left[(\mathbf{I} - \mathbf{Q}\mathbf{C}\mathbf{Q}^\top)\mathbf{B}(\mathbf{I} - \mathbf{Q}\mathbf{C}\mathbf{Q}^\top)\right]x \\
&= \sup_{(\mathbf{Q}^\top x)^\top\mathbf{\Lambda}(\mathbf{Q}^\top x)} (\mathbf{Q}^\top x)^\top\mathbb{E}\left[(\mathbf{I} - \mathbf{C})\mathbf{Q}^\top\mathbf{B}\mathbf{Q}(\mathbf{I} - \mathbf{C})\right](\mathbf{Q}^\top x) \\
&\overset{y=\mathbf{Q}^\top x}{=} \sup_{y^\top\mathbf{\Lambda}y=1} y^\top\mathbb{E}\left[(\mathbf{I} - \mathbf{C})\mathbf{\Lambda}(\mathbf{I} - \mathbf{C})\right]y \\
&= \sup_{y^\top\mathbf{\Lambda}y=1} (\mathbf{\Lambda}^{1/2}y)^\top\mathbb{E}\left[(\mathbf{I} - \mathbf{C})^2\right](\mathbf{\Lambda}^{1/2}y) \\
&\overset{z=\mathbf{\Lambda}^{1/2}y}{=} \sup_{\|z\|=1} z^\top \cdot \mathbf{Diag}(1-p) \cdot z \\
&= \max_{1\leq i\leq d}(1 - p_i) = 1 - \min_{1\leq i\leq d} p_i.
\end{aligned}$$

Hence

$$1 \leq \alpha(\mathcal{D}) + \mathbb{E}_{\mathcal{D}}\left[r/d\right] = 1 - \min_{1\leq i\leq d} p_i + \frac{1}{d}\sum_{i=1}^{d} p_i,$$

and equality occurs if and only if all probabilities p_i are equal to some $q \in [0,1]$. Thus, the optimal sketches are obtained by rotating the coordinate basis to the basis of eigenvectors of $\mathbf{Q}$ (i.e. $x \to \mathbf{Q}^\top x$), and then randomly sparsify coordinates with diagonal sketch matrix $\mathbf{C}$ (i.e. $\mathbf{Q}^\top x \to \mathbf{C}\mathbf{Q}^\top x = \mathbf{S}x$). We summarize this result in the following theorem.

Theorem 16. *Let $\mathbf{B} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ be the eigendecomposition of $\mathbf{B}$ of induced norm, $q \in [0,1]$ and $\mathbf{C}$ be random diagonal sketch corresponding to the random q-sparsifer. Then sketches $\mathbf{S} = \mathbf{C}\mathbf{Q}^\top$ are optimal with respect to variance against rank trade-off (39) with squared error $\alpha = 1 - q$ and expected rank $\mathbb{E}[r] = qd$.*

K.4 Random sketches with linear constraints

In this part we extend the theory of compressing vectors $x \in \mathbb{R}^d$ with an additional linear constraint $x \in \mathrm{Range}(\mathbf{A})$ for some $d \times d'$ matrix $\mathbf{A}$. Such scenarios occur when to-be-compressed vectors are the gradients of $f(w) = \phi(\mathbf{A}^\top w)$, for which $\nabla f(w) = \mathbf{A}\nabla\phi(\mathbf{A}^\top w) \in \mathrm{Range}(\mathbf{A})$. Without loss of generality, we may assume that $\mathbf{A}$ is of full column rank and consequently $d' = \dim \mathrm{Range}(\mathbf{A}) = \mathrm{rank}(\mathbf{A})$. The constraint $x \in \mathrm{Range}(\mathbf{A})$ then can be equivalently written as $x = \mathbf{A}x'$ for some $x' \in \mathbb{R}^{d'}$. The induced inner product and norm on $\mathrm{Range}(\mathbf{A})$ is then given by the matrix $\mathbf{A}^\top\mathbf{B}\mathbf{A}$ as

$$\langle x, y\rangle_{\mathbf{B}} = \langle \mathbf{A}x', \mathbf{A}y'\rangle_{\mathbf{B}} = \langle x', y'\rangle_{\mathbf{A}^\top\mathbf{B}\mathbf{A}}, \quad x = \mathbf{A}x',\, y = \mathbf{A}y'.$$

Notice that, since $\mathbf{S}x = \mathbf{S}\mathbf{A}x'$, communication of $x \in \mathbb{R}^d$ with sketches $\mathbf{S}$ reduces to communication of $x' \in \mathbb{R}^{d'}$ with sketches $\mathbf{S}\mathbf{A}$. Thus, the additional constraint $x \in \mathrm{Range}(\mathbf{A}) \subset \mathbb{R}^d$ reduces the problem to lower d'-dimension with sketches $\mathbf{S}\mathbf{A}, \mathbf{S} \sim \mathcal{D}$ and norm induced by $\mathbf{A}^\top\mathbf{B}\mathbf{A}$.

K.5 Variance against communication trade-off

The obtained lower bound (39) can be easily translated in terms of the number of bits. Assuming each float takes 32 bits to encode and there is no redundant row in $\mathbf{S}$ (i.e. $s = r$), then $\mathbf{S}x \in \mathbb{R}^r$ can be communicated with up to $b = 32r$ bits. Therefore, the lower bound (39) can be written as

$$\alpha + \frac{\mathbb{E}\left[b\right]}{32d} \geq 1, \tag{43}$$

which (ignoring the expectation) is exponentially stronger than the lower bound $\alpha \cdot 4^{b/d} \geq 1$ obtained for general compressors in (Safaryan et al., 2020). We visualize the comparison of these two lower bounds in Figure 6. Furthermore, denote by $\beta \coloneqq \mathbb{E}\left[b\right]/32d$ the expected communication reduction factor and recall that α is the portion of the expected lost of information. With this notation the above lower bound (43) turns to the following simple inequality

$$\alpha + \beta \geq 1,$$

showing the trade-off between information lost and communication reduction for linear compressors; namely more reduction in communication leads to bigger information loss and vice versa. In one extreme, when all $32d$ bits are sent, no reduction in communication is made ($\beta = 1$) and no information is lost ($\alpha = 0$). In other extreme, when no bits gets transferred ($\beta = 0$) we loose all information ($\alpha = 1$).

To conclude this section, let us investigate the optimality of random q-sparsifier with respect to the lower bound (43). Recall that random q-sparsifier is optimal with respect to (39). Let $q \in (0,1)$, and k be the (random) number of non-zero entries of sparsified vector. Clearly, $\mathbb{E}\left[k\right] = qd$ and to encode any k-sparse vector one needs $b = 32k + \log_2\binom{d}{k}$ bits. As we know from Theorem 16, the squared error $\alpha = 1 - q$. Therefore

$$\alpha + \beta = 1 - q + \tfrac{1}{32d}\mathbb{E}\left[32k + \log_2\tbinom{d}{k}\right] = 1 + \tfrac{1}{32d}\mathbb{E}\left[\log_2\tbinom{d}{k}\right] \leq 1 + \tfrac{1}{32}\mathbb{E}\left[H_2\left(\tfrac{k}{d}\right)\right] \leq 1 + \tfrac{H_2(q)}{32}.$$

The first inequality follows from the following estimate (only upper bound) for binomial coefficients

$$\frac{2^{dH_2(\tau)}}{\sqrt{8d\tau(1-\tau)}} \leq \binom{d}{\tau d} \leq \frac{2^{dH_2(\tau)}}{\sqrt{2\pi d\tau(1-\tau)}}, \quad 0 < \tau < 1,$$

where $H_2(\tau) = -\tau \log_2 \tau - (1-\tau)\log_2(1-\tau)$ is the binary entropy function in bits. The second inequality follows from concavity H_2 function and the Jensen's inequality. Because of the symmetry around $\tau = 1/2$ (namely $H_2(1-\tau) = H_2(\tau)$) and concavity of the function H_2, one can show that the maximum is achieved at $\tau = 1/2$ and $H_2(1/2) = 1$. Thus, in the worst case we have $\alpha + \beta \leq 33/32$ upper bound, when roughly half of the entries are chosen uniformly at random. For other values of q, it is even closer to the optimum; numerically $H_2(\tau) \approx (4\tau(1-\tau))^{3/4}$, $0 \leq \tau \leq 1$.

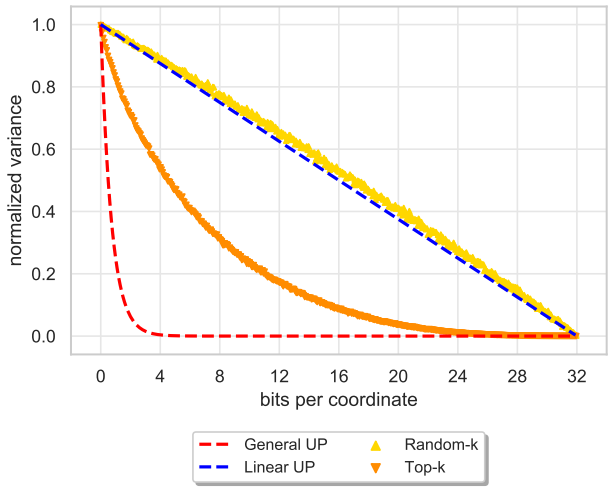

Figure 6: Comparison of general uncertainty principle $\alpha \cdot 4^{b/d} \geq 1$ (dashed red line) of Safaryan et al. (2020) against the new linear version (43) (dashed blue line). Each color represents one compression method: yellow for usual random sparsification with uniform probabilities and orange for greedy sparsification (a.k.a Top-k sparsification). Each triangle marker indicates one particular $d = 10^3$ dimensional vector randomly generated from Gaussian distribution, which subsequently gets compressed by the compression operator mentioned in the legend.

L PROOFS

L.1 PROOF OF THEOREM 9

Using smoothness of f, we have

$$\begin{aligned}\mathbb{E}f(x^{k+1}) &= \mathbb{E}f(x^k - \gamma \mathbf{C}\nabla f(x^k)) \\ &\leq f(x^k) - \gamma \left\langle \nabla f(x^k), \mathbb{E}\left[\mathbf{C}\nabla f(x^k)\right]\right\rangle + \frac{\gamma^2}{2}\mathbb{E}\left[\|\mathbf{C}\nabla f(x^k)\|^2_{\mathbf{L}}\right] \\ &= f(x^k) - \gamma\|\nabla f(x^k)\|^2 + \frac{\gamma^2}{2}\|\nabla f(x^k)\|^2_{\mathbb{E}[\mathbf{CLC}]} \\ &\leq f(x^k) - \gamma\left(2 - \gamma\lambda_{\max}\left(\mathbb{E}\left[\mathbf{CLC}\right]\right)\right)\cdot\frac{1}{2}\|\nabla f(x^k)\|^2.\end{aligned} \tag{44}$$

Computing the expectation inside, we get

$$\mathbb{E}\left[\mathbf{CLC}\right] = \mathbb{E}\left[(\mathrm{c}_i\mathrm{c}_j\mathbf{L}_{ij})_{i,j=1}^d\right] = \left(\frac{p_{ij}\mathbf{L}_{ij}}{p_ip_j}\right)_{i,j=1}^d = (\mathbf{Diag}(1/p)\mathbf{P}\mathbf{Diag}(1/p))\circ\mathbf{L} = \overline{\mathbf{P}}\circ\mathbf{L}. \tag{45}$$

Therefore, using the bound for the step size γ and strong convexity of f, we get

$$\begin{aligned}\mathbb{E}\left[f(x^{k+1}) - f(x^*)\right] &\leq \left(f(x^k) - f(x^*)\right) - \gamma\left(2 - \gamma\lambda_{\max}\left(\overline{\mathbf{P}}\circ\mathbf{L}\right)\right)\cdot\frac{1}{2}\|\nabla f(x^k)\|^2 \\ &\leq \left(f(x^k) - f(x^*)\right) - \frac{\gamma}{2}\|\nabla f(x^k)\|^2 \\ &\leq (1-\gamma\mu)\left(f(x^k) - f(x^*)\right),\end{aligned} \tag{46}$$

repeated application of which completes the proof.

L.2 PROOF OF THEOREM 13

The following lemmas will be useful to handle the computation with pseudo-inverses.

Lemma 17 (Lemma E.2 and E.3 (Hanzely & Richtárik, 2019b)). *If f is convex and $\mathbf{L}$-smooth, then for any $x, y \in \mathbb{R}^d$*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{1}{2}\|\nabla f(x) - \nabla f(y)\|^2_{\mathbf{L}^\dagger}. \tag{47}$$

If, in addition, f is bounded below, then $\nabla f(x) \in \mathrm{Range}(\mathbf{L}^\dagger) = \mathrm{Range}(\mathbf{L})$ for all $x \in \mathbb{R}^d$.

Lemma 18. *With $\overline{\mathbf{C}} = \mathbf{L}^{1/2}\mathbf{C}\mathbf{L}^{\dagger 1/2}$, the following holds*

$$\mathbb{E}\left[\mathbf{L}^{1/2}\left(\overline{\mathbf{C}} - \mathbf{I}\right)^\top\left(\overline{\mathbf{C}} - \mathbf{I}\right)\mathbf{L}^{1/2}\right] = \mathbf{L}^{1/2}\mathbf{L}^{\dagger 1/2}\left(\widetilde{\mathbf{P}}\circ\mathbf{L}\right)\mathbf{L}^{\dagger 1/2}\mathbf{L}^{1/2}. \tag{48}$$

Proof. Using the property $\mathbf{L}^{1/2}\mathbf{L}^{\dagger 1/2}\mathbf{L}^{1/2} = \mathbf{L}^{1/2}$ of pseudoinverse, we have

$$\begin{aligned}&\mathbb{E}\left[\mathbf{L}^{1/2}\left(\overline{\mathbf{C}} - \mathbf{I}\right)^\top\left(\overline{\mathbf{C}} - \mathbf{I}\right)\mathbf{L}^{1/2}\right] \\ =\ &\mathbb{E}\left[\mathbf{L}^{1/2}\left(\mathbf{L}^{\dagger 1/2}\mathbf{C}\mathbf{L}^{1/2} - \mathbf{I}\right)\left(\mathbf{L}^{1/2}\mathbf{C}\mathbf{L}^{\dagger 1/2} - \mathbf{I}\right)\mathbf{L}^{1/2}\right] \\ =\ &\mathbb{E}\left[\mathbf{L}^{1/2}\left(\mathbf{L}^{\dagger 1/2}\mathbf{C}\mathbf{L}\mathbf{C}\mathbf{L}^{\dagger 1/2} - \mathbf{L}^{\dagger 1/2}\mathbf{C}\mathbf{L}^{1/2} - \mathbf{L}^{1/2}\mathbf{C}\mathbf{L}^{\dagger 1/2} + \mathbf{I}\right)\mathbf{L}^{1/2}\right] \\ \overset{(45)}{=}\ &\mathbf{L}^{1/2}\left(\mathbf{L}^{\dagger 1/2}\left(\overline{\mathbf{P}}\circ\mathbf{L}\right)\mathbf{L}^{\dagger 1/2} - \mathbf{L}^{\dagger 1/2}\mathbf{L}^{1/2} - \mathbf{L}^{1/2}\mathbf{L}^{\dagger 1/2} + \mathbf{I}\right)\mathbf{L}^{1/2} \\ =\ &\mathbf{L}^{1/2}\left(\mathbf{L}^{\dagger 1/2}\left(\overline{\mathbf{P}}\circ\mathbf{L}\right)\mathbf{L}^{\dagger 1/2} - \mathbf{L}^{\dagger 1/2}\mathbf{L}\mathbf{L}^{\dagger 1/2}\right)\mathbf{L}^{1/2} \\ &\quad + \mathbf{L}^{1/2}\left(\mathbf{L}^{\dagger 1/2}\mathbf{L}\mathbf{L}^{\dagger 1/2} - \mathbf{L}^{\dagger^{1/2}}\mathbf{L}^{1/2} - \mathbf{L}^{1/2}\mathbf{L}^{\dagger^{1/2}} + \mathbf{I}\right)\mathbf{L}^{1/2} \\ =\ &\mathbf{L}^{1/2}\mathbf{L}^{\dagger 1/2}\left(\overline{\mathbf{P}}\circ\mathbf{L} - \mathbf{L}\right)\mathbf{L}^{\dagger 1/2}\mathbf{L}^{1/2} + \mathbf{L}^{1/2}\left(\mathbf{I} - \mathbf{L}^{\dagger 1/2}\mathbf{L}^{1/2}\right)\left(\mathbf{I} - \mathbf{L}^{1/2}\mathbf{L}^{\dagger 1/2}\right)\mathbf{L}^{1/2} \\ =\ &\mathbf{L}^{1/2}\mathbf{L}^{\dagger 1/2}\left(\widetilde{\mathbf{P}}\circ\mathbf{L}\right)\mathbf{L}^{\dagger 1/2}\mathbf{L}^{1/2} + \left(\mathbf{L}^{1/2} - \mathbf{L}^{1/2}\mathbf{L}^{\dagger 1/2}\mathbf{L}^{1/2}\right)\left(\mathbf{L}^{1/2} - \mathbf{L}^{1/2}\mathbf{L}^{\dagger 1/2}\mathbf{L}^{1/2}\right) \\ =\ &\mathbf{L}^{1/2}\mathbf{L}^{\dagger 1/2}\left(\widetilde{\mathbf{P}}\circ\mathbf{L}\right)\mathbf{L}^{\dagger 1/2}\mathbf{L}^{1/2}.\end{aligned}$$

□

For convenience we skip iteration count k, and write x, x^+ instead of x^k, x^{k+1}. Using non-expansiveness of the prox operator we get

$$\begin{aligned}
& \mathbb{E}\left[\|x^+ - x^*\|^2\right] \\
\leq\ & \mathbb{E}\left[\|x - x^* - \gamma\left(\mathbf{L}^{1/2}\mathbf{C}\mathbf{L}^{\dagger 1/2}\right)\nabla f(x) + \gamma\nabla f(x^*)\|^2\right] \\
=\ & \|x - x^*\|^2 - 2\gamma\langle x - x^*, \nabla f(x) - \nabla f(x^*)\rangle + \gamma^2\mathbb{E}\left[\|\left(\mathbf{L}^{1/2}\mathbf{C}\mathbf{L}^{\dagger 1/2}\right)\nabla f(x) - \nabla f(x^*)\|^2\right] \\
\leq\ & \|x - x^*\|^2 - 2\gamma\langle x - x^*, \nabla f(x) - \nabla f(x^*)\rangle \\
& +2\gamma^2\mathbb{E}\left[\|\mathbf{L}^{1/2}\mathbf{C}\mathbf{L}^{\dagger 1/2}\left(\nabla f(x) - \nabla f(x^*)\right)\|^2\right] + 2\gamma^2\mathbb{E}\left[\|\left(\mathbf{L}^{1/2}\mathbf{C}\mathbf{L}^{\dagger 1/2} - \mathbf{I}\right)\nabla f(x^*)\|^2\right] \\
\leq\ & \|x - x^*\|^2 - 2\gamma\langle x - x^*, \nabla f(x) - \nabla f(x^*)\rangle \\
& +2\gamma^2\lambda_{\max}(\mathbb{E}\left[\mathbf{C}\mathbf{L}\mathbf{C}\right])\|\mathbf{L}^{\dagger 1/2}\left(\nabla f(x) - \nabla f(x^*)\right)\|^2 + 2\gamma^2\mathbb{E}\left[\|\left(\mathbf{L}^{1/2}\mathbf{C}\mathbf{L}^{\dagger 1/2} - \mathbf{I}\right)\nabla f(x^*)\|^2\right] \\
\overset{(45),(49)}{\leq}\ & \|x - x^*\|^2 - 2\gamma\langle x - x^*, \nabla f(x) - \nabla f(x^*)\rangle \\
& +2\gamma^2\lambda_{\max}(\overline{\mathbf{P}}\circ\mathbf{L})\|\nabla f(x) - \nabla f(x^*)\|^2_{\mathbf{L}^\dagger} + 2\gamma^2\lambda_{\max}(\widetilde{\mathbf{P}}\circ\mathbf{L})\|\nabla f(x^*)\|^2_{\mathbf{L}^\dagger} \\
=\ & \|x - x^*\|^2 - 2\gamma\langle x - x^*, \nabla f(x) - \nabla f(x^*)\rangle + 2\gamma^2\widetilde{\mathcal{L}}\|\nabla f(x) - \nabla f(x^*)\|^2_{\mathbf{L}^\dagger} + 2\gamma^2\widetilde{\mathcal{L}}\|\nabla f(x^*)\|^2_{\mathbf{L}^\dagger},
\end{aligned}$$

where we used $\mathbb{E}\left[\mathbf{C}\mathbf{L}\mathbf{C}\right] = \overline{\mathbf{P}}\circ\mathbf{L}$ based on (45) and for the last term we used Lemma 17 to represent $\nabla f(x^*) = \mathbf{L}^{1/2}g_*$ and then applied Lemma 18

$$\begin{aligned}
\mathbb{E}\left[\left\|\left(\mathbf{L}^{1/2}\mathbf{C}\mathbf{L}^{\dagger 1/2} - \mathbf{I}\right)\nabla f(x^*)\right\|^2\right] &= \mathbb{E}\left[g_*^\top\mathbf{L}^{1/2}\left(\mathbf{L}^{\dagger 1/2}\mathbf{C}\mathbf{L}^{1/2} - \mathbf{I}\right)\left(\mathbf{L}^{1/2}\mathbf{C}\mathbf{L}^{\dagger 1/2} - \mathbf{I}\right)\mathbf{L}^{1/2}g_*\right] \\
&= \nabla f(x^*)^\top\left(\mathbf{L}^{\dagger 1/2}\left(\widetilde{\mathbf{P}}\circ\mathbf{L}\right)\mathbf{L}^{\dagger 1/2}\right)\nabla f(x^*) \\
&\leq \lambda_{\max}(\widetilde{\mathbf{P}}\circ\mathbf{L})\|\nabla f(x^*)\|^2_{\mathbf{L}^\dagger}.
\end{aligned} \tag{49}$$

Using the bound on step size $\gamma \leq 1/2\widetilde{c}$, strong convexity of f and (47), we continue as follows

$$\begin{aligned}
\mathbb{E}\left[\|x^+ - x^*\|^2\right] \quad \leq\quad & \|x - x^*\|^2 - \gamma\langle x - x^*, \nabla f(x) - \nabla f(x^*)\rangle \\
& -\gamma\left(\langle x - x^*, \nabla f(x) - \nabla f(x^*)\rangle - \|\nabla f(x) - \nabla f(x^*)\|^2_{\mathbf{L}^\dagger}\right) \\
& +2\gamma^2\widetilde{\mathcal{L}}\|\nabla f(x^*)\|^2_{\mathbf{L}^\dagger} \\
\overset{(47)}{\leq}\quad & (1 - \gamma\mu)\|x - x^*\|^2 + 2\gamma^2\widetilde{\mathcal{L}}\|\nabla f(x^*)\|^2_{\mathbf{L}^\dagger}.
\end{aligned}$$

Telescoping the above inequality, we complete the proof.

L.3 Proof of Theorem 3

In this proof we skip the iteration count k to simplify the notation. Define

$$\begin{aligned}
\mathbf{M}_i \quad &\coloneqq\quad \mathbf{L}_i^{1/2}\mathbb{E}\left[(\overline{\mathbf{C}}_i - \mathbf{I})^\top(\overline{\mathbf{C}}_i - \mathbf{I})\right]\mathbf{L}_i^{1/2} \\
&\overset{(48)}{=}\quad \mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}(\widetilde{\mathbf{P}}_i\circ\mathbf{L}_i)\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2} \\
&=\quad \mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}(\overline{\mathbf{P}}_i\circ\mathbf{L}_i - \mathbf{L}_i)\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2} \\
&=\quad \mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}(\overline{\mathbf{P}}_i\circ\mathbf{L}_i)\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2} - \mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2} \\
&=\quad \mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}(\overline{\mathbf{P}}_i\circ\mathbf{L}_i)\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2} - \mathbf{L}_i \\
&=\quad \mathbf{L}_i^{1/2}\left(\mathbb{E}\left[\overline{\mathbf{C}}_i^\top\overline{\mathbf{C}}_i\right] - \mathbf{I}\right)\mathbf{L}_i^{1/2}.
\end{aligned} \tag{50}$$

We are going to estimate the moment $\mathbb{E}\left[\|g(x)-\nabla f(x^*)\|^2\right]$ and show the following bound for the gradient estimator $g(x)=\frac{1}{n}\sum_{i=1}^n \overline{\mathbf{C}}_i\nabla f_i(x)$ (see line 5 of Algorithm 1):

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

Due to Lemma 17, we have $\nabla f_i(x)=\mathbf{L}_i^{1/2}r_i$ for some r_i. Therefore

$$\mathbb{E}\left[\overline{\mathbf{C}}_i\nabla f_i(x)\right]=\mathbb{E}\left[\mathbf{L}_i^{1/2}\mathbf{C}_i\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2}r_i\right]=\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2}r_i=\mathbf{L}_i^{1/2}r_i=\nabla f_i(x), \tag{51}$$

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

$$\begin{aligned}
\mathbb{E}\left[\|g(x)-\nabla f(x^*)\|^2\right] &= \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \overline{\mathbf{C}}_i\nabla f_i(x)-\nabla f(x^*)\right\|^2\right]\\
&= \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left[\left\|\overline{\mathbf{C}}_i\nabla f_i(x)-\nabla f(x^*)\right\|^2\right]+\frac{1}{n^2}\sum_{i\neq j}\mathbb{E}\left\langle \overline{\mathbf{C}}_i\nabla f_i(x)-\nabla f(x^*), \overline{\mathbf{C}}_j\nabla f_j(x)-\nabla f(x^*)\right\rangle\\
&= \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left[\left\|\overline{\mathbf{C}}_i\nabla f_i(x)\right\|^2\right]+\|\nabla f(x^*)\|^2-2\mathbb{E}\left\langle \overline{\mathbf{C}}_i\nabla f_i(x), \nabla f(x^*)\right\rangle\\
&\quad+\frac{1}{n^2}\sum_{i\neq j}\left\langle \nabla f_i(x)-\nabla f(x^*), \nabla f_j(x)-\nabla f(x^*)\right\rangle\\
&= \frac{1}{n^2}\sum_{i=1}^n \|\nabla f_i(x)\|^2_{\mathbb{E}\left[\overline{\mathbf{C}}_i^\top\overline{\mathbf{C}}_i\right]}+\|\nabla f(x^*)\|^2-2\left\langle \nabla f_i(x), \nabla f(x^*)\right\rangle\\
&\quad+\|\nabla f(x)-\nabla f(x^*)\|^2-\frac{1}{n^2}\sum_{i=1}^n\|\nabla f_i(x)-\nabla f(x^*)\|^2\\
&= \frac{1}{n^2}\sum_{i=1}^n \left\|\mathbf{L}_i^{1/2}r_i\right\|^2_{\mathbb{E}\left[\overline{\mathbf{C}}_i^\top\overline{\mathbf{C}}_i\right]-\mathbf{I}}+\frac{1}{n^2}\sum_{i=1}^n\|\nabla f_i(x)\|^2+\|\nabla f(x^*)\|^2-2\left\langle \nabla f_i(x), \nabla f(x^*)\right\rangle\\
&\quad+\|\nabla f(x)-\nabla f(x^*)\|^2-\frac{1}{n^2}\sum_{i=1}^n\|\nabla f_i(x)-\nabla f(x^*)\|^2\\
&= \|\nabla f(x)-\nabla f(x^*)\|^2+\frac{1}{n^2}\sum_{i=1}^n\|r_i\|^2_{\mathbf{M}_i}\\
&= \|\nabla f(x)-\nabla f(x^*)\|^2+\frac{1}{n^2}\sum_{i=1}^n\|r_i\|^2_{\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}(\widetilde{\mathbf{P}}_i\circ\mathbf{L}_i)\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2}}\\
&= \|\nabla f(x)-\nabla f(x^*)\|^2+\frac{1}{n^2}\sum_{i=1}^n\left\|\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x)\right\|^2_{\widetilde{\mathbf{P}}_i\circ\mathbf{L}_i},
\end{aligned}$$

which gives as the following decomposition

$$\mathbb{E}\left[\|g(x)-\nabla f(x^*)\|^2\right]=\|\nabla f(x)-\nabla f(x^*)\|^2+\frac{1}{n^2}\sum_{i=1}^n\left\|\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x)\right\|^2_{\widetilde{\mathbf{P}}_i\circ\mathbf{L}_i}. \tag{52}$$

For the first term it can be bounded using convexity and smoothness of f, namely $\|\nabla f(x) - \nabla f(x^*)\|^2 \leq 2LD_f(x, x^*)$. For the second term we proceed as follows

$$\begin{aligned}
\frac{1}{n^2}\sum_{i=1}^n \left\|\mathbf{L}_i^{\dagger^{1/2}}\nabla f_i(x)\right\|^2_{\widetilde{\mathbf{P}}_i\circ\mathbf{L}_i} &\leq \frac{1}{n^2}\sum_{i=1}^n \lambda_{\max}(\widetilde{\mathbf{P}}_i\circ\mathbf{L}_i)\|\mathbf{L}_i^{\dagger^{1/2}}\nabla f_i(x)\|^2 \\
&= \frac{1}{n^2}\sum_{i=1}^n \widetilde{\mathcal{L}}_i\|\nabla f_i(x)\|^2_{\mathbf{L}_i^\dagger} \\
&\leq \frac{2}{n^2}\sum_{i=1}^n \widetilde{\mathcal{L}}_i\|\nabla f_i(x) - \nabla f_i(x^*)\|^2_{\mathbf{L}_i^\dagger} + \frac{2}{n^2}\sum_{i=1}^n \widetilde{\mathcal{L}}_i\|\nabla f_i(x^*)\|^2_{\mathbf{L}_i^\dagger} \\
&\leq \frac{2}{n^2}\sum_{i=1}^n 2\widetilde{\mathcal{L}}_i D_{f_i}(x, x^*) + \frac{2\sigma^*}{n} \\
&= \frac{4\widetilde{\mathcal{L}}_{\max}}{n}D_f(x, x^*) + \frac{2\sigma^*}{n}.
\end{aligned} \tag{53}$$

Combining these two estimates, we get

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

It remains to apply the result of Gorbunov et al. (2020a).

L.4 Proof of Theorem 4

First, we show the unbiasedness of the estimator $g(x^k)$. In (51), we showed unbiasedness of $\overline{\mathbf{C}}_i^k\nabla f_i(x^k)$ using inclusion $\nabla f_i(x^k) \in \mathrm{Range}(\mathbf{L}_i)$. Assume for a moment that we also have $h_i^k \in \mathrm{Range}(\mathbf{L}_i)$. Hence, in the same way we can show $\mathbb{E}_k\left[\overline{\mathbf{C}}_i^k h_i^k\right] = h_i^k$, which implies the unbiasedness of g^k as

$$\mathbb{E}_k\left[g^k\right] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}_k\left[\overline{\mathbf{C}}_i^k\nabla f_i(x^k)\right] - \mathbb{E}_k\left[\overline{\mathbf{C}}_i^k h_i^k\right] + h_i^k = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x^k) = \nabla f(x^k).$$

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

$$
\begin{aligned}
\mathbb{E}\left[\|g^k - \nabla f(x^*)\|^2\right] &= \|\nabla f(x^k) - \nabla f(x^*)\|^2 + \mathbb{E}\left[\|g^k - \nabla f(x^k)\|^2\right] \\
&\leq 2LD_f(x^k, x^*) + \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \overline{\mathbf{C}}_i^k(\nabla f_i(x^k) - h_i^k) + h_i^k - \nabla f_i(x^k)\right\|^2\right] \\
&= 2LD_f(x^k, x^*) + \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left[\left\|(\overline{\mathbf{C}}_i^k - \mathbf{I})\mathbf{L}_i^{1/2} r_i^k\right\|^2\right] \\
&= 2LD_f(x^k, x^*) + \frac{1}{n^2}\sum_{i=1}^n \left\|r_i^k\right\|^2_{\mathbb{E}\left[\mathbf{L}_i^{1/2}(\overline{\mathbf{C}}_i^k - \mathbf{I})^\top(\overline{\mathbf{C}}_i^k - \mathbf{I})\mathbf{L}_i^{1/2}\right]} \\
&\overset{(50)}{=} 2LD_f(x^k, x^*) + \frac{1}{n^2}\sum_{i=1}^n \left\|r_i^k\right\|^2_{\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}(\widetilde{\mathbf{P}}_i \circ \mathbf{L}_i)\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2}} \\
&= 2LD_f(x^k, x^*) + \frac{1}{n^2}\sum_{i=1}^n \left\|\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x^k) - h_i^k)\right\|^2_{\widetilde{\mathbf{P}}_i \circ \mathbf{L}_i} \\
&\leq 2LD_f(x^k, x^*) + \frac{\widetilde{\mathcal{L}}_{\max}}{n^2}\sum_{i=1}^n \left\|\nabla f_i(x^k) - h_i^k\right\|^2_{\mathbf{L}_i^\dagger} \\
&\leq 2LD_f(x^k, x^*) + \frac{2\widetilde{\mathcal{L}}_{\max}}{n^2}\sum_{i=1}^n \left\|\nabla f_i(x^k) - f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger} + \frac{2\widetilde{\mathcal{L}}_{\max}}{n^2}\sum_{i=1}^n \left\|h_i^k - \nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger} \\
&\leq 2LD_f(x^k, x^*) + \frac{4\widetilde{\mathcal{L}}_{\max}}{n} D_f(x^k, x^*) + \frac{2\widetilde{\mathcal{L}}_{\max}}{n^2}\sum_{i=1}^n \left\|h_i^k - \nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger} \\
&= 2\left(L + \frac{2\widetilde{\mathcal{L}}_{\max}}{n}\right) D_f(x^k, x^*) + \frac{2\widetilde{\mathcal{L}}_{\max}}{n^2}\sum_{i=1}^n \left\|h_i^k - \nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger}
\end{aligned}
\tag{54}
$$

Then we deduce a recurrence relation for the last term $\sigma^k \coloneqq \frac{1}{n}\sum_{i=1}^n \left\|h_i^k - \nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger}$. For that we will need the following bounds

$$
0 \preceq \mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2} \preceq \mathbf{I}, \tag{55}
$$

which can be proved via SVD and eigenvalue decompositions. Since $\mathbf{L}_i$ is square, symmetric and positive semidefinite, we know that singular value decomposition and eigenvalue decompositions are the same. Let $\mathbf{L}_i^{1/2} = \mathbf{U}_i\mathbf{D}_i\mathbf{U}_i^\top$, where $\mathbf{D}_i$ is diagonal and $\mathbf{U}_i$ is orthogonal so that $\mathbf{U}_i^\top = \mathbf{U}_i^{-1}$. Then

$$
\mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2} = \mathbf{U}_i\mathbf{D}_i\mathbf{U}_i^\top\mathbf{U}_i\mathbf{D}_i^{\dagger 2}\mathbf{U}_i^\top\mathbf{U}_i\mathbf{D}_i\mathbf{U}_i^\top = \mathbf{U}_i\left(\mathbf{D}_i\mathbf{D}_i^{\dagger 2}\mathbf{D}_i\right)\mathbf{U}_i^\top = \mathbf{U}_i\left(\mathbf{D}_i\mathbf{D}_i^\dagger\right)\mathbf{U}_i^\top,
$$

which can admit eigenvalues only in $[0, 1]$ since the matrix $\mathbf{D}_i\mathbf{D}_i^\dagger$ is diagonal with entries either 0 or 1. Denote

$$
\omega_i = \lambda_{\max}\left(\mathbb{E}\left[(\mathbf{C}_i^k)^2\right]\right) - 1 = \max_{1 \leq j \leq d} \frac{1}{p_{i;j}} - 1. \tag{56}
$$

and bound each summand of σ^{k+1} as follows

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

where we used bounds $\alpha\le\frac{1}{1+\omega_i}$ and

$$\mathbb{E}\left[(\overline{\mathbf{C}}_i^k)^\top\mathbf{L}_i^\dagger\overline{\mathbf{C}}_i^k\right]=\mathbf{L}_i^{\dagger 1/2}\mathbb{E}\left[\mathbf{C}_i^k\mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2}\mathbf{C}_i^k\right]\mathbf{L}_i^{\dagger 1/2}\preceq\mathbf{L}_i^{\dagger 1/2}\mathbb{E}\left[(\mathbf{C}_i^k)^2\right]\mathbf{L}_i^{\dagger 1/2}.$$

Therefore

$$\begin{aligned}
\mathbb{E}_k\left[\sigma^{k+1}\right]&=\frac{1}{n}\sum_{i=1}^n\mathbb{E}_k\left[\left\|h_i^{k+1}-\nabla f_i(x^*)\right\|_{\mathbf{L}_i^\dagger}^2\right]\\
&\le\frac{1-\alpha}{n}\sum_{i=1}^n\left\|h_i^k-\nabla f_i(x^*)\right\|_{\mathbf{L}_i^\dagger}^2+\frac{\alpha}{n}\sum_{i=1}^n\left\|\nabla f_i(x^k)-\nabla f_i(x^*)\right\|_{\mathbf{L}_i^\dagger}^2\\
&\le(1-\alpha)\sigma^k+\frac{2\alpha}{n}\sum_{i=1}^n D_{f_i}(x^k,x^*)\\
&=(1-\alpha)\sigma^k+2\alpha D_f(x^k,x^*).
\end{aligned}$$

Thus, with $\alpha\le\frac{1}{1+\omega_{\max}}$, the estimator g^k of Algorithm 2 satisfies

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

It remains to apply Theorem 4.1 (Gorbunov et al., 2020a) with parameters $A=L+\frac{2}{n}\widetilde{\mathcal{L}}_{\max},\ B=\frac{2}{n}\widetilde{\mathcal{L}}_{\max},\ \rho=\alpha,\ C=\alpha$ and $M=\frac{4}{\alpha n}\widetilde{\mathcal{L}}_{\max},\ A+CM=L+\frac{6}{n}\widetilde{\mathcal{L}}_{\max},\ 1+\frac{B}{M}-\rho=1-\frac{\alpha}{2}$.

L.5 Proof of Theorem 5

Following the analysis of Li et al. (2020), define

$$Z^k\coloneqq\|z^k-x^*\|^2,\qquad Y^k\coloneqq F(y^k)-F(x^*),\qquad W^k\coloneqq F(w^k)-F(x^*),$$
$$H^k\coloneqq\frac{1}{n}\sum_{i=1}^n\|\nabla f_i(w^k)-h_i^k\|_{\mathbf{L}_i^\dagger}^2.$$

Lemma 19 (Lemma 2, (Li et al., 2020)). *Let* $\eta \leq \frac{1}{2L}$, $\theta_1 \leq \frac{1}{4}$, $\theta_2 = \frac{1}{2}$, $\gamma = \frac{\eta}{2(\theta_1+\eta\mu)}$ *and* $\beta = 1-\gamma\mu$. *Then*

$$\mathbb{E}\left[Z^{k+1}\right] + \frac{2\gamma\beta}{\theta_1}\mathbb{E}\left[Y^{k+1}\right] \leq \beta Z^k + (1-\theta_1-\theta_2)\frac{2\gamma\beta}{\theta_1}Y^k + 2\gamma\beta\frac{\theta_2}{\theta_1}W^k + \frac{\gamma\eta}{\theta_1}\mathbb{E}\left[\|g^k - \nabla f(x^k)\|^2\right] - \frac{\gamma}{4n\theta_1}\sum_{i=1}^n \|\nabla f_i(w^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} - \frac{\gamma}{8n\theta_1}\sum_{i=1}^n \|\nabla f_i(y^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger}.$$

Proof. Proof is the same as for the original lemma except we use $\mathbf{L}_i$-smoothness of f_i via (47).

$$f_i(u) \geq f_i(x^k) + \left\langle \nabla f_i(x^k), u - x^k \right\rangle + \frac{1}{2}\|\nabla f_i(u) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger}.$$

□

Lemma 20 (Lemma 3, (Li et al., 2020)).

$$\mathbb{E}\left[W^{k+1}\right] = (1-q)W^k + qY^k.$$

Lemma 21 (Lemma 4, (Li et al., 2020)).

$$\mathbb{E}\left[\|g^k - \nabla f(x^k)\|^2\right] \leq \frac{2\widetilde{\mathcal{L}}_{\max}}{n^2}\sum_{i=1}^n \|\nabla f_i(w^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} + \frac{2\widetilde{\mathcal{L}}_{\max}}{n}H^k.$$

Proof. Let $\nabla f_i(x^k) - h_i^k = \mathbf{L}_i^{1/2} r_i^k$. Then

$$\begin{aligned}
\mathbb{E}\left[\|g^k - \nabla f(x^k)\|^2\right] &= \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \overline{\mathbf{C}}_i^k(\nabla f_i(x^k) - h_i^k) - (\nabla f_i(x^k) - h_i^k)\right\|^2\right] \\
&= \frac{1}{n^2}\mathbb{E}\left[\left\|\sum_{i=1}^n (\overline{\mathbf{C}}_i^k - \mathbf{I})(\nabla f_i(x^k) - h_i^k)\right\|^2\right] = \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left[\left\|(\overline{\mathbf{C}}_i^k - \mathbf{I})\mathbf{L}_i^{1/2} r_i^k\right\|^2\right] \\
&= \frac{1}{n^2}\sum_{i=1}^n \left\|r_i^k\right\|^2_{\mathbf{L}_i^{1/2}\mathbb{E}\left[(\overline{\mathbf{C}}_i^k - \mathbf{I})^\top(\overline{\mathbf{C}}_i^k - \mathbf{I})\right]\mathbf{L}_i^{1/2}} \overset{(50)}{=} \frac{1}{n^2}\sum_{i=1}^n \left\|r_i^k\right\|^2_{\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}(\widetilde{\mathbf{P}}_i \circ \mathbf{L}_i)\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2}} \\
&= \frac{1}{n^2}\sum_{i=1}^n \left\|\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x^k) - h_i^k)\right\|^2_{\widetilde{\mathbf{P}}_i \circ \mathbf{L}_i} \leq \frac{\widetilde{\mathcal{L}}_{\max}}{n^2}\sum_{i=1}^n \left\|\nabla f_i(x^k) - h_i^k\right\|^2_{\mathbf{L}_i^\dagger} \\
&\leq \frac{2\widetilde{\mathcal{L}}_{\max}}{n^2}\sum_{i=1}^n \left\|\nabla f_i(x^k) - \nabla f_i(w^k)\right\|^2_{\mathbf{L}_i^\dagger} + \frac{2\widetilde{\mathcal{L}}_{\max}}{n^2}\sum_{i=1}^n \left\|\nabla f_i(w^k) - h_i^k\right\|^2_{\mathbf{L}_i^\dagger}.
\end{aligned}$$

□

Lemma 22 (Lemma 5, (Li et al., 2020)). *If* $\alpha \leq \frac{1}{1+\omega_{\max}}$, *where* $\omega_{\max} = \max_{1\leq i\leq n}\omega_i$ *and* $\omega_i = \max_{1\leq j\leq d}\frac{1}{p_{i;j}} - 1$, *then*

$$\mathbb{E}\left[H^{k+1}\right] \leq \left(1 - \frac{\alpha}{2}\right)H^k + \left(1 + \frac{2q}{\alpha}\right)\frac{2q}{n}\left(\sum_{i=1}^n \|\nabla f_i(w^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} + \sum_{i=1}^n \|\nabla f_i(w^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger}\right).$$

Proof. We start bounding the summands of H^{k+1}. Let $\nabla f_i(w^k) - h_i^k = \mathbf{L}_i^{1/2} r_i^k$.

$$
\begin{aligned}
\mathbb{E}_k\left[\left\|\nabla f_i(w^{k+1}) - h_i^{k+1}\right\|_{\mathbf{L}_i^\dagger}^2\right] &= q\mathbb{E}_k\left[\left\|\nabla f_i(y^k) - h_i^{k+1}\right\|_{\mathbf{L}_i^\dagger}^2\right] + (1-q)\mathbb{E}_k\left[\left\|\nabla f_i(w^k) - h_i^{k+1}\right\|_{\mathbf{L}_i^\dagger}^2\right] \\
&\leq q\left(1+\frac{2q}{\alpha}\right)\left\|\nabla f_i(w^k) - \nabla f_i(y^k)\right\|_{\mathbf{L}_i^\dagger}^2 + \left(1-q+\left(1+\frac{\alpha}{2q}\right)q\right)\mathbb{E}\left[\left\|\nabla f_i(w^k) - h_i^{k+1}\right\|_{\mathbf{L}_i^\dagger}^2\right] \\
&= q\left(1+\frac{2q}{\alpha}\right)\left\|\nabla f_i(w^k) - \nabla f_i(y^k)\right\|_{\mathbf{L}_i^\dagger}^2 + \left(1+\frac{\alpha}{2}\right)\mathbb{E}\left[\left\|\nabla f_i(w^k) - h_i^{k+1}\right\|_{\mathbf{L}_i^\dagger}^2\right] \\
&= q\left(1+\frac{2q}{\alpha}\right)\left\|\nabla f_i(w^k) - \nabla f_i(y^k)\right\|_{\mathbf{L}_i^\dagger}^2 + \left(1+\frac{\alpha}{2}\right)\mathbb{E}\left[\left\|(\mathrm{I}-\alpha\overline{\mathbf{C}}_i^k)(\nabla f_i(w^k) - h_i^k)\right\|_{\mathbf{L}_i^\dagger}^2\right] \\
&= q\left(1+\frac{2q}{\alpha}\right)\left\|\nabla f_i(w^k) - \nabla f_i(y^k)\right\|_{\mathbf{L}_i^\dagger}^2 + \left(1+\frac{\alpha}{2}\right)\left\|r_i^k\right\|_{\mathbf{L}_i^{1/2}\mathbb{E}\left[(\mathrm{I}-\alpha\overline{\mathbf{C}}_i^k)^\top\mathbf{L}_i^\dagger(\mathrm{I}-\alpha\overline{\mathbf{C}}_i^k)\right]\mathbf{L}_i^{1/2}}^2.
\end{aligned}
$$

Next, we simplify the matrix of the second term.

$$
\begin{aligned}
&\mathbf{L}_i^{1/2}\mathbb{E}\left[(\mathrm{I}-\alpha\overline{\mathbf{C}}_i^k)^\top\mathbf{L}_i^\dagger(\mathrm{I}-\alpha\overline{\mathbf{C}}_i^k)\right]\mathbf{L}_i^{1/2} \\
&= \mathbb{E}\left[\mathbf{L}_i^{1/2}(\mathrm{I}-\alpha\mathbf{L}_i^{\dagger 1/2}\mathbf{C}_i^k\mathbf{L}_i^{1/2})\mathbf{L}_i^\dagger(\mathrm{I}-\alpha\mathbf{L}_i^{1/2}\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2})\mathbf{L}_i^{1/2}\right] \\
&= \mathbb{E}\left[(\mathbf{L}_i^{1/2}-\alpha\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}\mathbf{C}_i^k\mathbf{L}_i^{1/2})\mathbf{L}_i^\dagger(\mathbf{L}_i^{1/2}-\alpha\mathbf{L}_i^{1/2}\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2})\right] \\
&= \mathbb{E}\Big[\mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2} - \alpha\mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2}\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2} \\
&\qquad\qquad - \alpha\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}\mathbf{C}_i^k\mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2} + \alpha^2\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}\mathbf{C}_i^k\mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2}\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2}\Big] \\
&\overset{(55)}{\preceq} \mathbb{E}\Big[\mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2} - \alpha\mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2}\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2} \\
&\qquad\qquad - \alpha\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}\mathbf{C}_i^k\mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2} + \alpha^2\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}(\mathbf{C}_i^k)^2\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2}\Big] \\
&= \mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2} - \alpha\mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2} - \alpha\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2} + \alpha^2\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}\mathbb{E}\left[(\mathbf{C}_i^k)^2\right]\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2} \\
&\overset{(56)}{\preceq} \mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2} - 2\alpha\mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2} + \alpha^2(\omega_i+1)\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2} \\
&= (1-2\alpha+\alpha^2(\omega_i+1))\mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2} \\
&\preceq (1-\alpha)\mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2},
\end{aligned}
$$

where in the last step we make use of the bound $\alpha \leq \frac{1}{1+\omega_{\max}} = \min_{1\leq i\leq n}\frac{1}{1+\omega_i}$. Then we finish the recurrence as follows

$$
\begin{aligned}
&\mathbb{E}_k\left[\left\|\nabla f_i(w^{k+1}) - h_i^{k+1}\right\|_{\mathbf{L}_i^\dagger}^2\right] \\
&\quad\leq q\left(1+\frac{2q}{\alpha}\right)\left\|\nabla f_i(w^k) - \nabla f_i(y^k)\right\|_{\mathbf{L}_i^\dagger}^2 + \left(1+\frac{\alpha}{2}\right)\left\|r_i^k\right\|_{\mathbf{L}_i^{1/2}\mathbb{E}\left[(\mathrm{I}-\alpha\overline{\mathbf{C}}_i^k)^\top\mathbf{L}_i^\dagger(\mathrm{I}-\alpha\overline{\mathbf{C}}_i^k)\right]\mathbf{L}_i^{1/2}}^2 \\
&\quad\leq q\left(1+\frac{2q}{\alpha}\right)\left\|\nabla f_i(w^k) - \nabla f_i(y^k)\right\|_{\mathbf{L}_i^\dagger}^2 + \left(1+\frac{\alpha}{2}\right)(1-\alpha)\left\|r_i^k\right\|_{\mathbf{L}_i^{1/2}\mathbf{L}_i^\dagger\mathbf{L}_i^{1/2}}^2 \\
&\quad= q\left(1+\frac{2q}{\alpha}\right)\left\|\nabla f_i(w^k) - \nabla f_i(y^k)\right\|_{\mathbf{L}_i^\dagger}^2 + \left(1+\frac{\alpha}{2}\right)(1-\alpha)\left\|\nabla f_i(w^k) - h_i^k\right\|_{\mathbf{L}_i^\dagger}^2 \\
&\quad\leq 2q\left(1+\frac{2q}{\alpha}\right)\left(\left\|\nabla f_i(w^k) - \nabla f_i(x^k)\right\|_{\mathbf{L}_i^\dagger}^2 + \left\|\nabla f_i(y^k) - \nabla f_i(x^k)\right\|_{\mathbf{L}_i^\dagger}^2\right) + \left(1-\frac{\alpha}{2}\right)\left\|\nabla f_i(w^k) - h_i^k\right\|_{\mathbf{L}_i^\dagger}^2.
\end{aligned}
$$

Averaging over $i \in [n]$ completes the proof. □

Proof of Theorem 5. Using the 4 lemmas above and $\theta_1 \leq \frac{1}{4},\ \theta_2 = \frac{1}{2}$, the Lyapunov function Ψ^{k+1} admits the following recurrence

$$
\begin{aligned}
\mathbb{E}\left[\Psi^{k+1}\right] &:= \mathbb{E}\left[Z^{k+1} + \frac{2\gamma\beta}{\theta_1}Y^{k+1} + 2\gamma\beta\frac{\theta_2(1+\theta_1)}{\theta_1 q}W^{k+1} + \frac{8\gamma\eta\widetilde{\mathcal{L}}_{\max}}{\alpha\theta_1 n}H^{k+1}\right] \\
&\overset{\text{Lemma 19}}{\leq} \beta Z^k + (1-\theta_1-\theta_2)\frac{2\gamma\beta}{\theta_1}Y^k + 2\gamma\beta\frac{\theta_2}{\theta_1}W^k + \frac{\gamma\eta}{\theta_1}\mathbb{E}\left[\|g^k - \nabla f(x^k)\|^2\right] \\
&\quad - \frac{\gamma}{4n\theta_1}\sum_{i=1}^n \|\nabla f_i(w^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} - \frac{\gamma}{8n\theta_1}\sum_{i=1}^n \|\nabla f_i(y^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} \\
&\quad + \mathbb{E}\left[2\gamma\beta\frac{\theta_2(1+\theta_1)}{\theta_1 q}W^{k+1} + \frac{8\gamma\eta\widetilde{\mathcal{L}}_{\max}}{\alpha\theta_1 n}H^{k+1}\right] \\
&\overset{\text{Lemma 20}}{=} \beta Z^k + (1-\theta_1-\theta_2)\frac{2\gamma\beta}{\theta_1}Y^k + 2\gamma\beta\frac{\theta_2}{\theta_1}W^k + \frac{\gamma\eta}{\theta_1}\mathbb{E}\left[\|g^k - \nabla f(x^k)\|^2\right] \\
&\quad - \frac{\gamma}{4n\theta_1}\sum_{i=1}^n \|\nabla f_i(w^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} - \frac{\gamma}{8n\theta_1}\sum_{i=1}^n \|\nabla f_i(y^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} \\
&\quad + 2\gamma\beta\frac{\theta_2(1+\theta_1)}{\theta_1 q}(1-q)W^k + 2\gamma\beta\frac{\theta_2(1+\theta_1)}{\theta_1}Y^k + \mathbb{E}\left[\frac{8\gamma\eta\widetilde{\mathcal{L}}_{\max}}{\alpha\theta_1 n}H^{k+1}\right] \\
&\leq \beta Z^k + \left(1-\frac{\theta_1}{2}\right)\frac{2\gamma\beta}{\theta_1}Y^k + \left(1-\frac{\theta_1 q}{2}\right)2\gamma\beta\frac{\theta_2(1+\theta_1)}{\theta_1 q}W^k \\
&\quad - \frac{\gamma}{4n\theta_1}\sum_{i=1}^n \|\nabla f_i(w^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} - \frac{\gamma}{8n\theta_1}\sum_{i=1}^n \|\nabla f_i(y^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} \\
&\quad + \frac{\gamma\eta}{\theta_1}\mathbb{E}\left[\|g^k - \nabla f(x^k)\|^2\right] + \mathbb{E}\left[\frac{8\gamma\eta\widetilde{\mathcal{L}}_{\max}}{\alpha\theta_1 n}H^{k+1}\right] \\
&\overset{\text{Lemma 21}}{\leq} \beta Z^k + \left(1-\frac{\theta_1}{2}\right)\frac{2\gamma\beta}{\theta_1}Y^k + \left(1-\frac{\theta_1 q}{2}\right)2\gamma\beta\frac{\theta_2(1+\theta_1)}{\theta_1 q}W^k \\
&\quad - \frac{\gamma}{4n\theta_1}\sum_{i=1}^n \|\nabla f_i(w^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} - \frac{\gamma}{8n\theta_1}\sum_{i=1}^n \|\nabla f_i(y^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} \\
&\quad + \frac{2\gamma\eta\widetilde{\mathcal{L}}_{\max}}{\theta_1 n^2}\|\nabla f_i(w^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} + \frac{2\gamma\eta\widetilde{\mathcal{L}}_{\max}}{\theta_1 n}H^k + \mathbb{E}\left[\frac{8\gamma\eta\widetilde{\mathcal{L}}_{\max}}{\alpha\theta_1 n}H^{k+1}\right] \\
&\overset{\text{Lemma 22}}{\leq} \beta Z^k + \left(1-\frac{\theta_1}{2}\right)\frac{2\gamma\beta}{\theta_1}Y^k + \left(1-\frac{\theta_1 q}{2}\right)2\gamma\beta\frac{\theta_2(1+\theta_1)}{\theta_1 q}W^k \\
&\quad - \frac{\gamma}{4n\theta_1}\sum_{i=1}^n \|\nabla f_i(w^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} - \frac{\gamma}{8n\theta_1}\sum_{i=1}^n \|\nabla f_i(y^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} \\
&\quad + \frac{2\gamma\eta\widetilde{\mathcal{L}}_{\max}}{\theta_1 n^2}\|\nabla f_i(w^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} + \frac{2\gamma\eta\widetilde{\mathcal{L}}_{\max}}{\theta_1 n}H^k + \frac{8\gamma\eta\widetilde{\mathcal{L}}_{\max}}{\alpha\theta_1 n}\left(1-\frac{\alpha}{2}\right)H^k \\
&\quad + \left(1+\frac{2q}{\alpha}\right)\frac{16\gamma\eta\widetilde{\mathcal{L}}_{\max}q}{\alpha\theta_1 n^2}\left(\sum_{i=1}^n \|\nabla f_i(w^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} + \sum_{i=1}^n \|\nabla f_i(y^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger}\right) \\
&= \beta Z^k + \left(1-\frac{\theta_1}{2}\right)\frac{2\gamma\beta}{\theta_1}Y^k + \left(1-\frac{\theta_1 q}{2}\right)2\gamma\beta\frac{\theta_2(1+\theta_1)}{\theta_1 q}W^k + \left(1-\frac{\alpha}{4}\right)\frac{8\gamma\eta\widetilde{\mathcal{L}}_{\max}}{\alpha\theta_1 n}H^k \\
&\quad - \frac{\gamma}{n\theta_1}\left(\frac{1}{8} - \frac{2\eta\widetilde{\mathcal{L}}_{\max}}{n}\right)\sum_{i=1}^n \|\nabla f_i(w^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} \\
&\quad - \frac{\gamma}{n\theta_1}\left(\frac{1}{8} - \left(1+\frac{2q}{\alpha}\right)\frac{16\eta\widetilde{\mathcal{L}}_{\max}q}{\alpha n}\right)\left(\sum_{i=1}^n \|\nabla f_i(w^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger} + \sum_{i=1}^n \|\nabla f_i(y^k) - \nabla f_i(x^k)\|^2_{\mathbf{L}_i^\dagger}\right).
\end{aligned}
$$

To make the last two lines disappear from the recurrence, we need to make sure

$$\frac{1}{8}-\frac{2\eta\widetilde{\mathcal{L}}_{\max}}{n}\geq 0 \qquad \text{and} \qquad \frac{1}{8}-\left(1+\frac{2q}{\alpha}\right)\frac{16\eta\widetilde{\mathcal{L}}_{\max}q}{\alpha n}\geq 0,$$

or equivalently

$$\eta\leq\frac{n}{16\widetilde{\mathcal{L}}_{\max}} \qquad \text{and} \qquad \eta\leq\frac{n}{64\widetilde{\mathcal{L}}_{\max}}\cdot\frac{1}{\frac{2q}{\alpha}\left(\frac{2q}{\alpha}+1\right)}.$$

Since $\alpha\leq\frac{1}{\omega_{\max}+1}$ (see Lemma 22) and we also need to have $\eta\leq\frac{1}{2L}$ (see Lemma 19), we can set

$$\eta=\min\left(\frac{1}{2L},\frac{n}{64\widetilde{\mathcal{L}}_{\max}\left(2q(\omega_{\max}+1)+1\right)^2}\right).$$

Therefore

$$\begin{aligned}\mathbb{E}\left[\Psi^{k+1}\right]&\leq\beta Z^k+\left(1-\frac{\theta_1}{2}\right)\frac{2\gamma\beta}{\theta_1}Y^k+\left(1-\frac{\theta_1 q}{2}\right)2\gamma\beta\frac{\theta_2(1+\theta_1)}{\theta_1 q}W^k+\left(1-\frac{\alpha}{4}\right)\frac{8\gamma\eta\widetilde{\mathcal{L}}_{\max}}{\alpha\theta_1 n}H^k\\&\leq\left(1-\frac{\eta\mu}{4\theta_1}\right)Z^k+\left(1-\frac{\theta_1}{2}\right)\frac{2\gamma\beta}{\theta_1}Y^k+\left(1-\frac{\theta_1 q}{2}\right)2\gamma\beta\frac{\theta_2(1+\theta_1)}{\theta_1 q}W^k+\left(1-\frac{\alpha}{4}\right)\frac{8\gamma\eta\widetilde{\mathcal{L}}_{\max}}{\alpha\theta_1 n}H^k\\&\leq\left(1-\min\left\{\frac{\alpha}{4},\frac{q}{8},\frac{\sqrt{\eta\mu q}}{4}\right\}\right)\Psi^k,\end{aligned}$$

where we set $\gamma=\frac{\eta}{2(\theta_1+\eta\mu)}$, $\beta=1-\gamma\mu\leq 1-\frac{\eta\mu}{4\theta_1}$ due to $\eta\mu\leq\theta_1$, and $\theta_1=\min\left\{\frac{1}{4},\sqrt{\frac{\eta\mu}{q}}\right\}$. After telescoping we get an ε-solution $\mathbb{E}\left[\|z^k-x^*\|^2\right]\leq\varepsilon$ after

$$\max\left(4(1+\omega_{\max}),\frac{8}{q},4\sqrt{\frac{2}{\mu q}\max\left(L,\frac{32\widetilde{\mathcal{L}}_{\max}\left(2q(\omega_{\max}+1)+1\right)^2}{n}\right)}\right)\log\frac{\Psi^0}{\varepsilon}$$

iterations. Choosing $q=\min\left\{1,\frac{\max\left(1,\sqrt{\frac{nL}{32\widetilde{\mathcal{L}}_{\max}}}-1\right)}{2(1+\omega_{\max})}\right\}$ we can simplify the above iteration complexity into

$$k=\begin{cases}\widetilde{\mathcal{O}}\left(\omega_{\max}+\sqrt{\frac{\widetilde{\mathcal{L}}_{\max}(1+\omega_{\max})}{\mu n}}\right) & \text{if} \quad nL\leq 128\widetilde{\mathcal{L}}_{\max}\\ \widetilde{\mathcal{O}}\left(1+\omega_{\max}+\sqrt{\frac{1+\omega_{\max}}{\sqrt{n}}\frac{\sqrt{\widetilde{\mathcal{L}}_{\max}L}}{\mu}}\right) & \text{if} \quad 128\widetilde{\mathcal{L}}_{\max}<nL\leq 32\widetilde{\mathcal{L}}_{\max}(2\omega_{\max}+3)^2\\ \widetilde{\mathcal{O}}\left(\omega_{\max}+\sqrt{\frac{L}{\mu}}\right) & \text{if} \quad 32\widetilde{\mathcal{L}}_{\max}(2\omega_{\max}+3)^2<nL.\end{cases}$$

Combining last two cases concludes the proof. □

M IMPROVEMENTS OVER THE ORIGINAL METHODS

In this part we provide detailed derivations skipped in Section E. Recall parameters ν, ν_s describing the distribution of matrices $\mathbf{L}_i$:

$$\nu \coloneqq \frac{\sum_{i=1}^n L_i}{\max\limits_{1\leq i\leq n} L_i},\ \nu_s \coloneqq \max_{1\leq i\leq n} \frac{\sum_{j=1}^d \mathbf{L}_{i;j}^{1/s}}{\max\limits_{1\leq j\leq d} \mathbf{L}_{i;j}^{1/s}}, \tag{57}$$

where $L_i = \lambda_{\max}(\mathbf{L}_i)$ and we will choose $s = 1$ or $s = 2$. Let $L_{\max} = \max_{1\leq i\leq n} L_i$.

M.1 IMPORTANCE SAMPLING FOR DCGD+

Let $\tau = \mathbb{E}\left[|S_i|\right] = \sum_{j=1}^d p_{i;j}$ be the expected mini-batch size for the samplings S_i. Notice that convergence rate of DCGD+ depends on $\widetilde{\mathcal{L}}_{\max} = \max_{1\leq i\leq n} \widetilde{\mathcal{L}}_i$. Since each node $i \in [n]$ generates its own diagonal sketch $\mathbf{C}_i$ independently from others, each node can optimize $\widetilde{\mathcal{L}}_i = \lambda_{\max}(\widetilde{\mathbf{P}}_i \circ \mathbf{L}_i)$ independently based on local smoothness matrix $\mathbf{L}_i$. In general, minimizing $\lambda_{\max}(\widetilde{\mathbf{P}}_i \circ \mathbf{L}_i)$ with respect to probability matrix $\widetilde{\mathbf{P}}_i$ is hard. However, we can find the optimal probabilities when each node generates via an independent sampling, namely $p_{i;jl} = p_{i;j}p_{i;l}$ if $j \neq l$. Then

$$\lambda_{\max}(\widetilde{\mathbf{P}}_i \circ \mathbf{L}_i) = \max_{1\leq j\leq d} \left(\frac{1}{p_{i;j}} - 1\right) \mathbf{L}_{i;j}, \tag{58}$$

for which we can find the optimal probabilities $p_{i;j}$. To minimize the maximum term in (58), we should have $\left({}^1/_{p_{i;j}} - 1\right) \mathbf{L}_{i;j} = \rho_i$ for some $\rho_i \geq 0$. Then the solution is

$$p_{i;j} = \frac{\mathbf{L}_{i;j}}{\mathbf{L}_{i;j} + \rho_i}, \tag{59}$$

where $\rho_i \geq 0$ is the unique solution to $\sum_{j=1}^d \frac{\mathbf{L}_{i;j}}{\mathbf{L}_{i;j}+\rho_i} = \tau$. The latter does not allow closed form solution for ρ_i, but it can be computed numerically using one dimensional solvers. Hence, we can efficiently compute the optimal probabilities (59). Moreover, we can deduce a simple upper bound for ρ_i

$$\tau = \sum_{j=1}^d \frac{\mathbf{L}_{i;j}}{\mathbf{L}_{i;j} + \rho_i} \leq \sum_{j=1}^d \frac{\mathbf{L}_{i;j}}{\rho_i} = \frac{1}{\rho_i} \sum_{j=1}^d \mathbf{L}_{i;j}, \tag{60}$$

which gives us an upper bound for $\widetilde{\mathcal{L}}_i$ as follows

$$\widetilde{\mathcal{L}}_i = \lambda_{\max}(\widetilde{\mathbf{P}}_i \circ \mathbf{L}_i) = \rho_i \leq \frac{1}{\tau} \sum_{j=1}^d \mathbf{L}_{i;j} \overset{(57)}{\leq} \frac{\nu_1}{\tau} \mathbf{L}_{\max}. \tag{61}$$

Proof of Remark 3. Using the following inequalities with respect to matrix order

$$\mathbf{L} \preceq \frac{1}{n} \sum_{i=1}^n \mathbf{L}_i, \quad \mathbf{L}_i \preceq n\mathbf{L}, \tag{62}$$

we bound L as follows

$$L = \lambda_{\max}\left(\mathbf{L}\right) \overset{(62)}{\leq} \lambda_{\max}\left(\frac{1}{n} \sum_{i=1}^n \mathbf{L}_i\right) \leq \frac{1}{n} \sum_{i=1}^n \lambda_{\max}\left(\mathbf{L}_i\right) = \frac{1}{n} \sum_{i=1}^n L_i \overset{(57)}{\leq} \frac{\nu}{n} L_{\max}. \tag{63}$$

Fix $\tau = \sum_{j=1}^d p_{i;j} \in [0, d]$ expected mini-batch of coordinates for all nodes $i \in [n]$. Then, with probabilities (59) we have

$$\frac{\widetilde{\mathcal{L}}_{\max}}{n} = \frac{1}{n} \max_{1\leq i\leq n} \widetilde{\mathcal{L}}_i = \frac{1}{n} \max_{1\leq i\leq n} \rho_i \overset{(61)}{\leq} \frac{\nu_1}{\tau n} \mathbf{L}_{\max} \leq \frac{\nu_1}{\tau n} L_{\max},$$

To get it upper bounded by $L_{\max}$, notice that $\max_{1\leq j\leq d}\mathbf{L}_{i;j}\leq\lambda_{\max}(\mathbf{L}_i)=L_i$, which implies

$$\mathbf{L}_{\max}=\max_{1\leq i\leq n}\max_{1\leq j\leq d}\mathbf{L}_{i;j}\leq\max_{1\leq i\leq n}L_i=L_{\max}. \tag{64}$$

Therefore

$$L+\frac{\widetilde{\mathcal{L}}_{\max}}{n}\leq\left(\frac{\nu}{n}+\frac{\nu_1}{\tau n}\right)L_{\max}.$$

□

M.2 Importance sampling for DIANA+

To find optimal probabilities for DIANA+, we minimize $\omega_{\max}+\frac{\widetilde{\mathcal{L}}_{\max}}{\mu n}$ part of the complexity (19) when each node uses an independent sampling as for DCGD+. Definitions of $\widetilde{\mathcal{L}}_{\max}$ and $\omega_{\max}$ imply

$$\omega_{\max}+\frac{\widetilde{\mathcal{L}}_{\max}}{\mu n}=\max_{ij}\left(\frac{1}{p_{i;j}}-1\right)+\max_{ij}\left(\frac{1}{p_{i;j}}-1\right)\frac{\mathbf{L}_{i;j}}{\mu n}=\Theta\left(\max_{ij}\left(\frac{1}{p_{i;j}}-1\right)\left(\frac{\mathbf{L}_{i;j}}{\mu n}+1\right)\right). \tag{65}$$

Therefore it is equivalent to minimize the following for each node $i\in[n]$ independently:

$$\max_{1\leq j\leq d}\left(\frac{1}{p_{i;j}}-1\right)\mathbf{L}'_{i;j},\quad \mathbf{L}'_{i;j}\coloneqq\frac{\mathbf{L}_{i;j}}{\mu n}+1\geq 1, \tag{66}$$

This can be solved in the same way as (58). The optimal probabilities are

$$p_{i;j}=\frac{\mathbf{L}'_{i;j}}{\mathbf{L}'_{i;j}+\rho'_i}=\frac{\frac{\mathbf{L}_{i;j}}{\mu n}+1}{\frac{\mathbf{L}_{i;j}}{\mu n}+1+\rho'_i} \tag{67}$$

and an upper bound for ρ'_i is analogous to (61)

$$\rho'_i\leq\frac{1}{\tau}\sum_{j=1}^{d}\mathbf{L}'_{i;j}=\frac{1}{\tau}\sum_{j=1}^{d}\left(\frac{\mathbf{L}_{i;j}}{\mu n}+1\right)=\frac{d}{\tau}+\frac{1}{n\tau}\sum_{j=1}^{d}\frac{\mathbf{L}_{i;j}}{\mu}\overset{(57)}{\leq}\frac{d}{\tau}+\frac{\nu_1}{n\tau}\frac{\mathbf{L}_{\max}}{\mu}\overset{(64)}{\leq}\frac{d}{\tau}+\frac{\nu_1}{n\tau}\frac{L_{\max}}{\mu}. \tag{68}$$

Proof of Remark 4. With probabilities (67) we can upper bound the complexity (19) as follows

$$\begin{aligned}\omega_{\max}+\frac{\widetilde{\mathcal{L}}_{\max}}{\mu n}&\overset{(65)}{\leq}2\max_{1\leq i\leq n}\max_{1\leq j\leq d}\left(\frac{1}{p_{i;j}}-1\right)\mathbf{L}'_{i;j}\\&\overset{(67)}{=}\frac{2}{\tau}\max_{1\leq i\leq n}\rho'_i\\&\overset{(68)}{\leq}\frac{2d}{\tau}+\frac{2\nu_1}{\tau n}\frac{L_{\max}}{\mu}.\end{aligned} \tag{69}$$

Combined with (63), we have

$$\omega_{\max}+\frac{L}{\mu}+\frac{\widetilde{\mathcal{L}}_{\max}}{\mu n}\leq\frac{2d}{\tau}+\left(\frac{\nu}{n}+\frac{2\nu_1}{\tau n}\right)\frac{L_{\max}}{\mu}.$$

□

Remark 7 (Improvement over standard DGD)**.** *Let us estimate how much improvement do we get with respect to standard Distributed Gradient Descent (DGD), where each node computes full gradients $\nabla f_i(x^k)$ and sends dense updates to the server in each iteration. The iteration complexity of DGD is $\widetilde{\mathcal{O}}(\frac{L}{\mu})$. To compare it against the complexity (19) of DIANA+ we use the same setup as in previous remarks (namely, independent samplings with probabilities (26) and $\tau=d/n$). Since $\mathbf{L}_i\preceq n\mathbf{L}$, we have $L_{\max}=\max_{i\in[n]}\lambda_{\max}(\mathbf{L}_i)\leq nL$. Hence, (27) implies*

$$\omega_{\max}+\frac{L}{\mu}+\frac{\widetilde{\mathcal{L}}_{\max}}{\mu n}\leq 2n+\frac{3nL}{\mu},$$

which is $\mathcal{O}(n)$ times bigger than the iteration complexity of DGD. However, in case of DGD, each node sends n times more bits to the server. In total, DIANA+ and DGD have the same communication complexity in the worst case. To illustrate the best complexity DIANA+ can provide, consider the special case when $\mathbf{L}_i = \mathbf{L}$ for all $i \in [n]$ and $\nu_1 = \mathcal{O}(1)$. Then, clearly $L_{\max} = L$ and we get $\widetilde{\mathcal{O}}(n + \frac{L}{\mu})$ complexity for DIANA+, yielding up to n times speedup against DGD. Moreover, in case of diagonal matrices $\mathbf{L}_i$, DIANA+ spends n times less local computation on partial derivatives and guarantees additional n times speedup.

M.3 Independent sampling for ADIANA+

For the accelerated method ADIANA+, we construct probabilities $p_{i;j}$ similar to (59) and (67) as follows

$$p_{i;j} \coloneqq \left(\frac{\mathbf{L}'_{i;j}}{\mathbf{L}'_{i;j} + \rho''_i}\right)^{1/2} = \left(\frac{\frac{\mathbf{L}_{i;j}}{\mu n} + 1}{\frac{\mathbf{L}_{i;j}}{\mu n} + 1 + \rho''_i}\right)^{1/2}, \quad \mathbf{L}'_{i;j} = \frac{\mathbf{L}_{i;j}}{\mu n} + 1 \geq 1, \tag{70}$$

where ρ''_i is determined uniquely from $\sum_{j=1}^d \left(\frac{\mathbf{L}'_{i;j}}{\mathbf{L}'_{i;j}+\rho''_i}\right)^{1/2} = \tau$. Notice that

$$\tau = \sum_{j=1}^d \left(\frac{\mathbf{L}'_{i;j}}{\mathbf{L}'_{i;j} + \rho''_i}\right)^{1/2} \leq \sum_{j=1}^d \left(\frac{\mathbf{L}'_{i;j}}{\rho''_i}\right)^{1/2} = \frac{1}{\sqrt{\rho''_i}} \sum_{j=1}^d \sqrt{\mathbf{L}'_{i;j}}.$$

Therefore

$$\begin{aligned}\sqrt{\rho''_i} &\leq \frac{1}{\tau}\sum_{j=1}^d \sqrt{\frac{\mathbf{L}_{i;j}}{\mu n} + 1} \leq \frac{1}{\tau}\sum_{j=1}^d \left(\sqrt{\frac{\mathbf{L}_{i;j}}{\mu n}} + 1\right) \leq \frac{d}{\tau} + \frac{1}{\tau}\sum_{j=1}^d \sqrt{\frac{\mathbf{L}_{i;j}}{\mu n}} \\ &\overset{(57)}{\leq} \frac{d}{\tau} + \frac{\nu_2}{\tau}\sqrt{\frac{\mathbf{L}_{\max}}{\mu n}} \overset{(64)}{\leq} \frac{d}{\tau} + \frac{\nu_2}{\tau}\sqrt{\frac{L_{\max}}{\mu n}}\end{aligned} \tag{71}$$

Proof of Remark 5. We bound terms $\omega_{\max}$ and $\frac{\mathcal{L}_{\max}}{\mu n}$ using probabilities (70) as follows:

$$\omega_{\max} = \max_{i,j}\left(\frac{1}{p_{i;j}} - 1\right) = \max_{i,j}\left(\sqrt{\frac{\rho''_i}{\mathbf{L}'_{i;j}} + 1} - 1\right) \leq \max_{i,j}\sqrt{\frac{\rho''_i}{\mathbf{L}'_{i;j}}} \overset{(70)}{\leq} \max_{i,j}\sqrt{\rho''_i} \overset{(71)}{\leq} \frac{d}{\tau} + \frac{\nu_2}{\tau}\sqrt{\frac{L_{\max}}{\mu n}}. \tag{72}$$

$$\frac{\mathcal{L}_{\max}}{\mu n} \overset{(58)}{=} \max_{i,j}\left(\frac{1}{p_{i;j}} - 1\right)\frac{\mathbf{L}_{i;j}}{\mu n} \overset{(70)}{\leq} \max_{i,j}\frac{\sqrt{\rho''_i}\frac{\mathbf{L}_{i;j}}{\mu n}}{\sqrt{\frac{\mathbf{L}_{i;j}}{\mu n} + 1}} \leq \max_{i,j}\sqrt{\rho''_i}\sqrt{\frac{\mathbf{L}_{i;j}}{\mu n}} \overset{(71)}{\leq} \left(\frac{d}{\tau} + \frac{\nu_2}{\tau}\sqrt{\frac{L_{\max}}{\mu n}}\right)\sqrt{\frac{L_{\max}}{\mu n}}. \tag{73}$$

Let ν and ν_2 are $\mathcal{O}(1)$. Denote $\omega = \frac{d}{\tau}$, $\kappa_i = \frac{L_i}{\mu}$ and $\kappa_{\max} = \max_{i\in[n]}\kappa_i$. Then with this notation we have

$$\begin{aligned}\frac{L}{\mu} &\leq \frac{\nu}{n}\kappa_{\max} = \mathcal{O}\left(\frac{\kappa_{\max}}{n}\right) \\ \omega_{\max} &\leq \omega + \frac{\nu_2}{\tau}\sqrt{\frac{\kappa_{\max}}{n}} = \omega\left(1 + \frac{\nu_2}{d}\sqrt{\frac{\kappa_{\max}}{n}}\right) = \mathcal{O}\left(\omega\left(1 + \frac{\sqrt{\kappa_{\max}}}{d\sqrt{n}}\right)\right) \\ \frac{\mathcal{L}_{\max}}{\mu n} &\leq \left(\omega + \frac{\nu_2}{\tau}\sqrt{\frac{\kappa_{\max}}{n}}\right)\sqrt{\frac{\kappa_{\max}}{n}} = \mathcal{O}\left(\omega\left(1 + \frac{\sqrt{\kappa_{\max}}}{d\sqrt{n}}\right)\frac{\sqrt{\kappa_{\max}}}{\sqrt{n}}\right)\end{aligned} \tag{74}$$

Then, in case of $nL \leq \widetilde{\mathcal{L}}_{\max}$, we have

$$\omega_{\max} + \sqrt{\omega_{\max}\frac{\widetilde{\mathcal{L}}_{\max}}{\mu n}} = \mathcal{O}\left(\omega\left(1 + \frac{\sqrt{\kappa_{\max}}}{d\sqrt{n}}\right)\left(1 + \left(\frac{\kappa_{\max}}{n}\right)^{1/4}\right)\right),$$

which should be compared with $\mathcal{O}\left(\omega\left(1+\sqrt{\frac{\kappa_{\max}}{n}}\right)\right)$ (Li et al., 2020). If $\kappa_{\max}=\mathcal{O}(nd^2)$, then we get $\mathcal{O}(\sqrt{d})$ speedup factor. If $nL>\widetilde{\mathcal{L}}_{\max}$, then

$$
\begin{aligned}
&\omega_{\max}+\sqrt{\frac{L}{\mu}}+\sqrt{\omega_{\max}\sqrt{\frac{\widetilde{\mathcal{L}}_{\max}}{\mu n}}\sqrt{\frac{L}{\mu}}} \\
&\qquad=\mathcal{O}\left(\omega\left(1+\frac{\sqrt{\kappa_{\max}}}{d\sqrt{n}}\right)+\sqrt{\frac{\kappa_{\max}}{n}}+\sqrt{\omega\left(1+\frac{\sqrt{\kappa_{\max}}}{d\sqrt{n}}\right)\sqrt{\frac{\kappa_{\max}}{n}}\sqrt{\omega\left(1+\frac{\sqrt{\kappa_{\max}}}{d\sqrt{n}}\right)\sqrt{\frac{\kappa_{\max}}{n}}}}\right) \\
&\qquad=\mathcal{O}\left(\omega\left(1+\frac{\sqrt{\kappa_{\max}}}{d\sqrt{n}}\right)+\sqrt{\frac{\kappa_{\max}}{n}}+\left[\omega\left(1+\frac{\sqrt{\kappa_{\max}}}{d\sqrt{n}}\right)\sqrt{\frac{\kappa_{\max}}{n}}\right]^{3/4}\right),
\end{aligned}
$$

which should be compared with $\omega+\kappa_{\max}+\omega^{3/4}n^{1/4}\sqrt{\frac{\kappa_{\max}}{n}}$ (Li et al., 2020). If $\kappa_{\max}=\mathcal{O}(nd^2)$, then we get $\mathcal{O}(\sqrt{n})$ times smaller second term and $\mathcal{O}\left((nd)^{1/4}\right)$ times smaller third term.

□

N VARIANCE REDUCTION: ISEGA+

In this part we apply our redesign to another variance reduced method called ISEGA (Mishchenko et al., 2020; Hanzely & Richtárik, 2019b). At the core of ISEGA, the mechanism for variance reduction is based on SEGA method (Hanzely et al., 2018). The key difference between ISEGA and DIANA is that ISEGA updates the control variates h more aggressively using projection instead of the mere α-step towards the projection used in DIANA. Adapting our matrix-smoothness-aware sparsification to ISEGA, we define the update rule of control vectors h_i^k as follows (**for now assume $\mathbf{L}_i$ is invertible**)

$$\begin{aligned}
h_i^{k+1} = & \underset{\substack{h\in \mathrm{Range}(\mathbf{L}_i) \\ \mathbf{C}_i^k \mathbf{L}_i^{\dagger^{1/2}} \nabla f_i(x^k) = \mathbf{C}_i^k \mathbf{L}_i^{\dagger^{1/2}} h}}{\arg\min} \|h - h_i^k\|^2_{\mathbf{L}_i^\dagger} \\
&= h_i^k + \mathbf{L}_i \mathbf{L}_i^{\dagger^{1/2}} \mathbf{C}_i^k \left(\mathbf{C}_i^k \mathbf{L}_i^{\dagger^{1/2}} \mathbf{L}_i \mathbf{L}_i^{\dagger^{1/2}} \mathbf{C}_i^k \right)^\dagger \mathbf{C}_i^k \mathbf{L}_i^{\dagger^{1/2}} (\nabla f_i(x^k) - h_i^k) \\
&= h_i^k + \mathbf{L}_i^{1/2} \mathbf{C}_i^k \left(\mathbf{C}_i^k \mathbf{C}_i^k \right)^\dagger \mathbf{C}_i^k \mathbf{L}_i^{\dagger^{1/2}} (\nabla f_i(x^k) - h_i^k) \\
&= h_i^k + \mathbf{L}_i^{1/2} \mathbf{Diag}(\mathbf{P}_i) \mathbf{C}_i^k \mathbf{L}_i^{\dagger^{1/2}} (\nabla f_i(x^k) - h_i^k).
\end{aligned}$$

Note that the update rule in DIANA+ has the form

$$h_i^{k+1} = h_i^k + \alpha \mathbf{L}_i^{1/2} \mathbf{C}_i^k \mathbf{L}_i^{\dagger^{1/2}} (\nabla f_i(x^k) - h_i^k)$$

for some fixed scalar $\alpha > 0$, and thus is more conservative. Note that we choose the gradient estimator to be the same $g_i^k = h_i^k + \mathbf{L}_i^{1/2} \mathbf{C}_i^k \mathbf{L}_i^{\dagger^{1/2}} (\nabla f_i(x^k) - h_i^k)$. The method is presented as Algorithm 7.

Algorithm 7 ISEGA+

Input: Initial point $x^0 \in \mathbb{R}^d$, initial shifts $h_i^0 \in \mathbb{R}^d$, current point x^k, step size parameter γ and α, sketch $\mathbf{C}_i^k$ and $\overline{\mathbf{C}}_i^k := \mathbf{L}_i^{1/2} \mathbf{C}_i^k \mathbf{L}_i^{\dagger^{1/2}}$, current shifts $h_1^k, \ldots, h_n^k$ and $h^k := \frac{1}{n} \sum_{i=1}^n h_i^k$.
on each node
  get x^k from the server
  send sparse update $\Delta_i^k = \mathbf{C}_i^k \mathbf{L}_i^{\dagger^{1/2}} (\nabla f_i(x^k) - h_i^k)$
  $g_i^k = h_i^k + \mathbf{L}_i^{1/2} \Delta_i^k$
  $h_i^{k+1} = h_i^k + \mathbf{L}_i^{1/2} \mathbf{Diag}(\mathbf{P}_i) \Delta_i^k$
on server
  get sparse updates Δ_i^k from each node
  $g^k = \frac{1}{n} \sum_{i=1}^n g_i^k = h^k + \frac{1}{n} \sum_{i=1}^n \mathbf{L}_i^{1/2} \Delta_i^k$
  $x^{k+1} = \mathrm{prox}_{\gamma R}(x^k - \gamma g^k)$
  $h^{k+1} = \frac{1}{n} \sum_{i=1}^n h_i^{k+1} = h^k + \frac{1}{n} \sum_{i=1}^n \mathbf{L}_i^{1/2} \mathbf{Diag}(\mathbf{P}_i) \Delta_i^k$

Note that we can not obtain the convergence rate of ISEGA+ directly from the framework of Gorbunov et al. (2020a). Instead, to get the tight convergence rate, we shall cast it as an instance of GJS method (Hanzely & Richtárik, 2019b). Theorem 23 provides the result – we can see that the worst case complexity is identical to DIANA+. In terms of the practical performance, we expect ISEGA+ to outperform DIANA+ due to the more aggressive update rule of control variates.

Theorem 23. *Suppose that* $\gamma \leq \frac{1}{\frac{4\widetilde{\mathcal{L}}_{\max}}{n} + 2L + \mu(\omega_{\max}+1)}$. *Then, we have*

$$\mathbb{E}[\Psi^k] \leq (1 - \gamma\mu)\Psi^0,$$

where

$$\Psi^k := \|x^k - x^*\|^2 + \frac{\gamma}{2n} \sum_{i=1}^n \|\phi_i^k - \mathbf{L}_i^{\dagger^{1/2}} \nabla f_i(x^*)\|^2_{\mathbf{Diag}(\mathbf{P}_i)^{-1}}$$

and $\phi_i^k := \mathbf{L}_i^{\dagger^{1/2}} h_i^k$. *Consequently, the overall complexity of ISEGA+ is*

$$\tilde{\mathcal{O}} \left(\omega_{\max} + \frac{L}{\mu} + \frac{\widetilde{\mathcal{L}}_{\max}}{n\mu} \right).$$

Proof. The proof can be seen as a special case of the generalized Jacobian sketching theory of Hanzely & Richtárik (2019b). For the sake of clarity, we provide a specialized proof here.

Note first that by (54),we have

$$\mathbb{E}\left[\|g^k - \nabla f(x^*)\|^2\right] \leq 2\left(L + \frac{2\widetilde{\mathcal{L}}_{\max}}{n}\right) D_f(x^k, x^*) + \frac{2\widetilde{\mathcal{L}}_{\max}}{n^2}\sum_{i=1}^n \left\|\phi_i^k - \mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)\right\|^2.$$

Similarly, we have

$$\begin{aligned}
& \mathbb{E}\left[\|\phi_i^{k+1} - \mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)\|^2_{\mathbf{Diag}(\mathbf{P}_i)^{-1}}\right] \\
= \;& \mathbb{E}\left[\|\phi_i^k + \mathbf{Diag}(\mathbf{P}_i)\mathbf{C}_i^k(\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^k) - \phi_i^k) - \mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)\|^2_{\mathbf{Diag}(\mathbf{P}_i)^{-1}}\right] \\
= \;& \mathbb{E}\left[\|(\mathbf{I} - \mathbf{Diag}(\mathbf{P}_i)\mathbf{C}_i^k)(\phi_i^k - \mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)) + \mathbf{Diag}(\mathbf{P}_i)\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x^k) - \nabla f_i(x^*))\|^2_{\mathbf{Diag}(\mathbf{P}_i)^{-1}}\right] \\
= \;& \mathbb{E}\left[\|(\mathbf{I} - \mathbf{Diag}(\mathbf{P}_i)\mathbf{C}_i^k)\mathbf{Diag}(\mathbf{P}_i)^{-\frac{1}{2}}(\phi_i^k - \mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)) + \mathbf{Diag}(\mathbf{P}_i)^{1/2}\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x^k) - \nabla f_i(x^*))\|^2\right] \\
= \;& \mathbb{E}\left[\|(\mathbf{I} - \mathbf{Diag}(\mathbf{P}_i)\mathbf{C}_i^k)\mathbf{Diag}(\mathbf{P}_i)^{-\frac{1}{2}}(\phi_i^k - \mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*))\|^2\right] + \mathbb{E}\left[\|\mathbf{Diag}(\mathbf{P}_i)^{1/2}\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x^k) - \nabla f_i(x^*))\|^2\right] \\
= \;& \|\phi_i^k - \mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)\|^2_{\mathbf{Diag}(\mathbf{P}_i)^{-1}-\mathbf{I}} + \|\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x^k) - \nabla f_i(x^*))\|^2 \\
\leq \;& \|\phi_i^k - \mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)\|^2_{\mathbf{Diag}(\mathbf{P}_i)^{-1}-\mathbf{I}} + 2D_{f_i}(x^k, x^*)
\end{aligned}$$

and therefore

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \|\phi_i^{k+1} - \mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)\|^2_{\mathbf{Diag}(\mathbf{P}_i)^{-1}}\right] \leq \frac{1}{n}\sum_{i=1}^n \|\phi_i^k - \mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)\|^2_{\mathbf{Diag}(\mathbf{P}_i)^{-1}-\mathbf{I}} + 2D_f(x^k, x^*) \tag{75}$$

Following the classical analysis of SGD (i.e., proof of Lemma C.1 of Gorbunov et al. (2020a)), we get

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

Adding $\frac{\gamma}{2}$-multiple of (75) to the above, we get

$$\begin{aligned}
&\mathbb{E}\left[\|x^{k+1} - x^*\|^2\right] + \frac{\gamma}{2}\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \|\phi_i^{k+1} - \mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)\|^2_{\mathbf{Diag}(\mathbf{P}_i)^{-1}}\right] \\
&\quad \leq (1-\gamma\mu)\|x^k - x^*\|^2 - 2\gamma\left(\frac{1}{2} - \gamma\left(L + \frac{2\widetilde{\mathcal{L}}_{\max}}{n}\right)\right) D_f(x^k, x^*) \\
&\qquad + \frac{2\widetilde{\mathcal{L}}_{\max}\gamma^2}{n^2}\sum_{i=1}^n \left\|\phi_i^k - \mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)\right\|^2 + \frac{\gamma}{2n}\sum_{i=1}^n \|\phi_i^k - \mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)\|^2_{\mathbf{Diag}(\mathbf{P}_i)^{-1}-\mathbf{I}}
\end{aligned} \tag{76}$$

Next, note that we have

$$\begin{aligned}
&\frac{2\widetilde{\mathcal{L}}_{\max}\gamma^2}{n^2}\sum_{i=1}^n \left\|\phi_i^k - \mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)\right\|^2 + \frac{\gamma}{2n}\sum_{i=1}^n \|\phi_i^k - \mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)\|^2_{\mathbf{Diag}(\mathbf{P}_i)^{-1}-\mathbf{I}} \\
&\qquad \leq \frac{(1-\gamma\mu)\gamma}{2n}\sum_{i=1}^n \|\phi_i^k - \mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)\|^2_{\mathbf{Diag}(\mathbf{P}_i)^{-1}}
\end{aligned} \tag{77}$$

since it is equivalent to

$$\frac{4\widetilde{\mathcal{L}}_{\max}\gamma}{n}\sum_{i=1}^{n}\left\|\phi_i^k-\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)\right\|^2+\gamma\mu\sum_{i=1}^{n}\|\phi_i^k-\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)\|^2_{\mathbf{Diag}(\mathbf{P}_i)^{-1}}\leq\sum_{i=1}^{n}\|\phi_i^k-\mathbf{L}_i^{\dagger 1/2}\nabla f_i(x^*)\|^2,$$

which holds since $\gamma\leq\frac{1}{\frac{4\widetilde{\mathcal{L}}_{\max}}{n}+\mu(\omega_{\max}+1)}$.

To finish the proof, it remains to plug (77) into (76), use that $\gamma\leq\frac{1}{\frac{4\widetilde{\mathcal{L}}_{\max}}{n}+2L}$ and unroll the recurrence.

□

O VARIANCE REDUCTION WITH BI-DIRECTIONAL COMPRESSION: DIANA++

In this method, the master server applies compression in its turn with sketch $\mathbf{C}$ independently. Thus, we maintain an additional control vector H^k, which helps to reduce the variance coming from the master's sparsification. Moreover, nodes keep track of H^k just like the central server.

Algorithm 8 DIANA++

Input: Initial point $x^0 \in \mathbb{R}^d$, initial shifts $h_i^0 \in \text{Range}(\mathbf{L}_i),\ H^0 \in \text{Range}(\mathbf{L})$, current point x^k, step size parameter γ, α and β, sketch $\mathbf{C}_i^k$ and $\overline{\mathbf{C}}_i^k \coloneqq \mathbf{L}_i^{1/2}\mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}$, current shifts $h_1^k, \ldots, h_n^k, H^k$ and $h^k \coloneqq \frac{1}{n}\sum_{i=1}^n h_i^k$.
on each node
  send sparse update $\Delta_i^k = \mathbf{C}_i^k\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x^k) - h_i^k)$
  $\overline{\Delta}_i^k = \mathbf{L}_i^{1/2}\Delta_i^k,\ g_i^k = h_i^k + \overline{\Delta}_i^k, h_i^{k+1} = h_i^k + \alpha\overline{\Delta}_i^k$
on server
  get sparse updates Δ_i^k from each node
  $\overline{\Delta}^k = \frac{1}{n}\sum_{i=1}^n \overline{\Delta}_i^k = \frac{1}{n}\sum_{i=1}^n \mathbf{L}_i^{1/2}\Delta_i^k$
  $g^k = \overline{\Delta}^k + h^k = \frac{1}{n}\sum_{i=1}^n \overline{\mathbf{C}}_i^k\left(\nabla f_i(x^k) - h_i^k\right) + h_i^k$
  send sparse update $\delta^k = \mathbf{C}^k\mathbf{L}^{\dagger 1/2}(g^k - H^k)$
  $\bar{\delta}^k = \mathbf{L}^{1/2}\delta^k,\ \hat{g}^k = H^k + \bar{\delta}^k = H^k + \overline{\mathbf{C}}^k\left(g^k - H^k\right)$
  $x^{k+1} = \text{prox}_{\gamma R}(x^k - \gamma\hat{g}^k)$
  $h^{k+1} = h^k + \alpha\overline{\Delta}^k$
  $H^{k+1} = H^k + \beta\bar{\delta}^k$
on each node
  get δ^k from the server
  reconstruct $\bar{\delta}^k = \mathbf{L}^{1/2}\delta^k,\ \hat{g}^k = H^k + \bar{\delta}^k = H^k + \overline{\mathbf{C}}^k\left(g^k - H^k\right)$
  $x^{k+1} = \text{prox}_{\gamma R}(x^k - \gamma\hat{g}^k)$
  $H^{k+1} = H^k + \beta\bar{\delta}^k$

Theorem 24. *Let Assumptions 2 and 3 hold and assume that each node generates its own diagonal sketch $\mathbf{C}_i$ independently from others. The master server, in its turn, generates $\mathbf{C}$ independently from the nodes. Then, Algorithm 8 has the following iteration complexity*

$$\mathcal{O}\left(\frac{1}{\min\left(\alpha - \beta\theta', \beta\right)} + \frac{\alpha + \beta\theta + \beta\theta'}{\min\left(\alpha - \beta\theta', \beta\right)}\left(\frac{L}{\mu} + \frac{\widetilde{\mathcal{L}}}{\mu} + \frac{\widetilde{\mathcal{L}}\widetilde{\mathcal{L}}'_{\max}}{n\mu} + \frac{\widetilde{\mathcal{L}}_{\max}}{n\mu}\right)\right),$$

where we made the following notations

$$\theta \coloneqq \frac{n\widetilde{\mathcal{L}}}{\widetilde{\mathcal{L}}_{\max} + 2\widetilde{\mathcal{L}}\widetilde{\mathcal{L}}'_{\max}} \leq \frac{n}{2\widetilde{\mathcal{L}}'_{\max}}, \quad \theta' \coloneqq \frac{2\theta}{n}\widetilde{\mathcal{L}}'_{\max} \leq 1 \in [0,1]$$
$$\widetilde{\mathcal{L}}'_{\max} \coloneqq \max_{1\leq i\leq n} \lambda_{\max}\left(\widetilde{\mathbf{P}}_i \circ (\mathbf{L}_i^{1/2}\mathbf{L}^{\dagger}\mathbf{L}_i^{1/2})\right), \quad \widetilde{\mathcal{L}} \coloneqq \lambda_{\max}\left(\widetilde{\mathbf{P}} \circ \mathbf{L}\right)$$

with bounds $\alpha \leq \frac{1}{1+\omega_{\max}} = \max_{i\in[n]}\max_{j\in[d]}\frac{1}{p_{i;j}}$ and $\beta \leq \frac{1}{1+\omega} = \max_{j\in[d]}\frac{1}{p_j}$.

Remark 8. *Note that, when master does not compress the messages, then we have $\widetilde{\mathbf{P}} = \mathbf{0}$. This implies the same complexity we had for DIANA+ as quantities $\widetilde{\mathcal{L}},\ \theta,\ \theta'$ are all become zeros.*

Proof. The proof follows the same structure as for DIANA+, with additional variance reduction process introduced for the master server. Analogously, we start bounding the following second moment:

$$\mathbb{E}\left[\|\hat{g}^k - \nabla f(x^*)\|^2\right] = \mathbb{E}\left[\|\hat{g}^k - g^k\|^2\right] + \mathbb{E}\left[\|g^k - \nabla f(x^*)\|^2\right]. \tag{78}$$

We can bound the second term as it was done in (54):

$$\mathbb{E}\left[\|g^k - \nabla f(x^*)\|^2\right] \leq 2\left(L + \frac{2\widetilde{\mathcal{L}}_{\max}}{n}\right)D_f(x^k, x^*) + \frac{2\widetilde{\mathcal{L}}_{\max}}{n^2}\sum_{i=1}^n \left\|h_i^k - \nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^{\dagger}}.$$

Then we decompose the first term $\mathbb{E}\left[\|\hat{g}^k - g^k\|^2\right]$ into two as follows:

$$\begin{aligned}\mathbb{E}\left[\|\hat{g}^k - g^k\|^2\right] &= \mathbb{E}\left[\|\overline{\mathbf{C}}^k(g^k - H^k) - (g^k - H^k)\|^2\right] \\ &= \|g^k - H^k\|^2_{\mathbb{E}\left[(\mathbf{I}-\overline{\mathbf{C}}^k)^\top(\mathbf{I}-\overline{\mathbf{C}}^k)\right]} \\ &= \|g^k - H^k\|^2_{\mathbf{L}^{\dagger 1/2}(\widetilde{\mathbf{P}}\circ\mathbf{L})\mathbf{L}^{\dagger 1/2}} \\ &\leq \widetilde{\mathcal{L}}\|g^k - H^k\|^2_{\mathbf{L}^\dagger} \\ &\leq 2\widetilde{\mathcal{L}}\|g^k - \nabla f(x^*)\|^2_{\mathbf{L}^\dagger} + 2\widetilde{\mathcal{L}}\|H^k - \nabla f(x^*)\|^2_{\mathbf{L}^\dagger}.\end{aligned} \tag{79}$$

To bound each of the two summands in (79), we derive the analogue of (48).

$$\begin{aligned}&\mathbb{E}\left[\mathbf{L}_i^{1/2}\left(\overline{\mathbf{C}}_i - \mathbf{I}\right)^\top \mathbf{L}^\dagger \left(\overline{\mathbf{C}}_i - \mathbf{I}\right)\mathbf{L}_i^{1/2}\right] \\ &= \mathbb{E}\left[\mathbf{L}_i^{1/2}\left(\mathbf{L}_i^{\dagger 1/2}\mathbf{C}_i\mathbf{L}_i^{1/2} - \mathbf{I}\right)\mathbf{L}^\dagger\left(\mathbf{L}_i^{1/2}\mathbf{C}_i\mathbf{L}_i^{\dagger 1/2} - \mathbf{I}\right)\mathbf{L}_i^{1/2}\right] \\ &= \mathbb{E}\left[\mathbf{L}_i^{1/2}\left(\mathbf{L}_i^{\dagger 1/2}\mathbf{C}_i(\mathbf{L}_i^{1/2}\mathbf{L}^\dagger\mathbf{L}_i^{1/2})\mathbf{C}_i\mathbf{L}_i^{\dagger 1/2} - \mathbf{L}_i^{\dagger 1/2}\mathbf{C}_i\mathbf{L}_i^{1/2}\mathbf{L}^\dagger - \mathbf{L}^\dagger\mathbf{L}_i^{1/2}\mathbf{C}_i\mathbf{L}_i^{\dagger 1/2} + \mathbf{L}^\dagger\right)\mathbf{L}_i^{1/2}\right] \\ &\overset{(45)}{=} \mathbf{L}_i^{1/2}\left(\mathbf{L}_i^{\dagger 1/2}\left(\overline{\mathbf{P}}_i \circ (\mathbf{L}_i^{1/2}\mathbf{L}^\dagger\mathbf{L}_i^{1/2})\right)\mathbf{L}_i^{\dagger 1/2} - \mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2}\mathbf{L}^\dagger - \mathbf{L}^\dagger\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2} + \mathbf{L}^\dagger\right)\mathbf{L}_i^{1/2} \\ &= \mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}\left(\overline{\mathbf{P}}_i \circ (\mathbf{L}_i^{1/2}\mathbf{L}^\dagger\mathbf{L}_i^{1/2})\right)\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2} - \mathbf{L}_i^{1/2}\mathbf{L}^\dagger\mathbf{L}_i^{1/2} \\ &= \mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}\left(\widetilde{\mathbf{P}}_i \circ (\mathbf{L}_i^{1/2}\mathbf{L}^\dagger\mathbf{L}_i^{1/2})\right)\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2}.\end{aligned} \tag{80}$$

Then we bound them as follows. First, we have

$$\begin{aligned}\mathbb{E}\left[\|g^k - \nabla f(x^*)\|^2_{\mathbf{L}^\dagger}\right] &= \|\nabla f(x^k) - \nabla f(x^*)\|^2_{\mathbf{L}^\dagger} + \mathbb{E}\left[\|g^k - \nabla f(x^k)\|^2_{\mathbf{L}^\dagger}\right] \\ &\leq 2D_f(x^k, x^*) + \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \overline{\mathbf{C}}_i^k(\nabla f_i(x^k) - h_i^k) + h_i^k - \nabla f_i(x^k)\right\|^2_{\mathbf{L}^\dagger}\right] \\ &= 2D_f(x^k, x^*) + \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left[\left\|(\overline{\mathbf{C}}_i^k - \mathbf{I})\mathbf{L}_i^{1/2} r_i^k\right\|^2_{\mathbf{L}^\dagger}\right] \\ &= 2D_f(x^k, x^*) + \frac{1}{n^2}\sum_{i=1}^n \left\|r_i^k\right\|^2_{\mathbb{E}\left[\mathbf{L}_i^{1/2}(\overline{\mathbf{C}}_i^k-\mathbf{I})^\top\mathbf{L}^\dagger(\overline{\mathbf{C}}_i^k-\mathbf{I})\mathbf{L}_i^{1/2}\right]} \\ &\overset{(80)}{=} 2D_f(x^k, x^*) + \frac{1}{n^2}\sum_{i=1}^n \left\|r_i^k\right\|^2_{\mathbf{L}_i^{1/2}\mathbf{L}_i^{\dagger 1/2}(\widetilde{\mathbf{P}}_i\circ(\mathbf{L}_i^{1/2}\mathbf{L}^\dagger\mathbf{L}_i^{1/2}))\mathbf{L}_i^{\dagger 1/2}\mathbf{L}_i^{1/2}} \\ &= 2D_f(x^k, x^*) + \frac{1}{n^2}\sum_{i=1}^n \left\|\mathbf{L}_i^{\dagger 1/2}(\nabla f_i(x^k) - h_i^k)\right\|^2_{\widetilde{\mathbf{P}}_i\circ(\mathbf{L}_i^{1/2}\mathbf{L}^\dagger\mathbf{L}_i^{1/2})} \\ &\leq 2D_f(x^k, x^*) + \frac{\widetilde{\mathcal{L}}'_{\max}}{n^2}\sum_{i=1}^n \left\|\nabla f_i(x^k) - h_i^k\right\|^2_{\mathbf{L}_i^\dagger} \\ &\leq 2D_f(x^k, x^*) + \frac{2\widetilde{\mathcal{L}}'_{\max}}{n^2}\sum_{i=1}^n \left\|\nabla f_i(x^k) - f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger} + \frac{2\widetilde{\mathcal{L}}'_{\max}}{n^2}\sum_{i=1}^n \left\|h_i^k - \nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger} \\ &\leq 2D_f(x^k, x^*) + \frac{4\widetilde{\mathcal{L}}'_{\max}}{n}D_f(x^k, x^*) + \frac{2\widetilde{\mathcal{L}}'_{\max}}{n^2}\sum_{i=1}^n \left\|h_i^k - \nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger} \\ &= 2\left(1 + \frac{2\widetilde{\mathcal{L}}'_{\max}}{n}\right)D_f(x^k, x^*) + \frac{2\widetilde{\mathcal{L}}'_{\max}}{n^2}\sum_{i=1}^n \left\|h_i^k - \nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger}\end{aligned} \tag{81}$$

Then, for the control vectors H^k at the master, we have

$$
\begin{aligned}
&\mathbb{E}_k\left[\left\|H^{k+1}-\nabla f(x^*)\right\|_{\mathbf{L}^\dagger}^2\right]\\
&\quad=\mathbb{E}_k\left[\left\|H^{k}-\nabla f(x^*)+\beta\bar{\delta}^k\right\|_{\mathbf{L}^\dagger}^2\right]\\
&\quad=\left\|H^{k}-\nabla f(x^*)\right\|_{\mathbf{L}^\dagger}^2+2\beta\mathbb{E}\left[\left\langle H^k-\nabla f(x^*),g^k-H^k\right\rangle_{\mathbf{L}^\dagger}\right]+\beta^2\mathbb{E}_k\left[\left\|\overline{\mathbf{C}}^k(g^k-H^k)\right\|_{\mathbf{L}^\dagger}^2\right]\\
&\quad=\left\|H^{k}-\nabla f(x^*)\right\|_{\mathbf{L}^\dagger}^2+2\beta\mathbb{E}_k\left[\left\langle H^k-\nabla f(x^*),g^k-H^k\right\rangle_{\mathbf{L}^\dagger}\right]+\beta^2\mathbb{E}_k\left[\left\|g^k-H^k\right\|_{\mathbb{E}[(\overline{\mathbf{C}}^k)^\top\mathbf{L}^\dagger\overline{\mathbf{C}}^k]}^2\right]\\
&\quad\leq\left\|H^{k}-\nabla f(x^*)\right\|_{\mathbf{L}^\dagger}^2+2\beta\mathbb{E}_k\left[\left\langle H^k-\nabla f(x^*),g^k-H^k\right\rangle_{\mathbf{L}^\dagger}\right]+\beta^2\mathbb{E}_k\left[\left\|g^k-H^k\right\|_{\mathbf{L}^{\dagger 1/2}\mathbb{E}[(\mathbf{C}^k)^2]\mathbf{L}^{\dagger 1/2}}^2\right]\\
&\quad\leq\left\|H^{k}-\nabla f(x^*)\right\|_{\mathbf{L}^\dagger}^2+2\beta\mathbb{E}_k\left[\left\langle H^k-\nabla f(x^*),g^k-H^k\right\rangle_{\mathbf{L}^\dagger}\right]+\beta^2(1+\omega)\mathbb{E}_k\left[\left\|g^k-H^k\right\|_{\mathbf{L}^\dagger}^2\right]\\
&\quad\leq\left\|H^{k}-\nabla f(x^*)\right\|_{\mathbf{L}^\dagger}^2+2\beta\mathbb{E}_k\left[\left\langle H^k-\nabla f(x^*),g^k-H^k\right\rangle_{\mathbf{L}^\dagger}\right]+\beta\mathbb{E}_k\left[\left\|g^k-H^k\right\|_{\mathbf{L}^\dagger}^2\right]\\
&\quad=(1-\beta)\left\|H^{k}-\nabla f(x^*)\right\|_{\mathbf{L}^\dagger}^2+\beta\mathbb{E}_k\left[\left\|g^k-\nabla f(x^*)\right\|_{\mathbf{L}^\dagger}^2\right]\\
&\quad\leq(1-\beta)\left\|H^{k}-\nabla f(x^*)\right\|_{\mathbf{L}^\dagger}^2+2\beta\left(1+\frac{2\widetilde{\mathcal{L}}'_{\max}}{n}\right)D_f(x^k,x^*)+\frac{2\beta\widetilde{\mathcal{L}}'_{\max}}{n^2}\sum_{i=1}^n\left\|h_i^k-\nabla f_i(x^*)\right\|_{\mathbf{L}_i^\dagger}^2
\end{aligned}
$$

Now, for some θ (to be defined later), let

$$
\sigma^k\coloneqq\frac{1}{n}\sum_{i=1}^n\|h_i^k-\nabla f_i(x^*)\|_{\mathbf{L}_i^\dagger}^2+\theta\|H^k-\nabla f(x^*)\|_{\mathbf{L}^\dagger}^2.
$$

Then, we have

$$
\begin{aligned}
&\mathbb{E}\left[\|\hat{g}^k-\nabla f(x^*)\|^2\right]\\
&\overset{(78)}{=}\mathbb{E}\left[\|\hat{g}^k-g^k\|^2\right]+\mathbb{E}\left[\|g^k-\nabla f(x^*)\|^2\right]\\
&\overset{(79)}{\leq}2\widetilde{\mathcal{L}}\mathbb{E}\left[\|g^k-\nabla f(x^*)\|_{\mathbf{L}^\dagger}^2\right]+2\widetilde{\mathcal{L}}\|H^k-\nabla f(x^*)\|_{\mathbf{L}^\dagger}^2+\mathbb{E}\left[\|g^k-\nabla f(x^*)\|^2\right]\\
&\overset{(81)}{\leq}4\widetilde{\mathcal{L}}\left(1+\frac{2\widetilde{\mathcal{L}}'_{\max}}{n}\right)D_f(x^k,x^*)+\frac{4\widetilde{\mathcal{L}}\widetilde{\mathcal{L}}'_{\max}}{n^2}\sum_{i=1}^n\left\|h_i^k-\nabla f_i(x^*)\right\|_{\mathbf{L}_i^\dagger}^2\\
&\quad+2\left(L+\frac{2\widetilde{\mathcal{L}}_{\max}}{n}\right)D_f(x^k,x^*)+\frac{2\widetilde{\mathcal{L}}_{\max}}{n^2}\sum_{i=1}^n\left\|h_i^k-\nabla f_i(x^*)\right\|_{\mathbf{L}_i^\dagger}^2\\
&\quad+2\widetilde{\mathcal{L}}\|H^k-\nabla f(x^*)\|_{\mathbf{L}^\dagger}^2\\
&=2\left(L+2\widetilde{\mathcal{L}}+\frac{4\widetilde{\mathcal{L}}\widetilde{\mathcal{L}}'_{\max}}{n}+\frac{2\widetilde{\mathcal{L}}_{\max}}{n}\right)D_f(x^k,x^*)\\
&\quad+\left(\frac{4\widetilde{\mathcal{L}}\widetilde{\mathcal{L}}'_{\max}}{n}+\frac{2\widetilde{\mathcal{L}}_{\max}}{n}\right)\frac{1}{n}\sum_{i=1}^n\left\|h_i^k-\nabla f_i(x^*)\right\|_{\mathbf{L}_i^\dagger}^2+2\widetilde{\mathcal{L}}\|H^k-\nabla f(x^*)\|_{\mathbf{L}^\dagger}^2\\
&=2\left(L+2\widetilde{\mathcal{L}}+\frac{4\widetilde{\mathcal{L}}\widetilde{\mathcal{L}}'_{\max}}{n}+\frac{2\widetilde{\mathcal{L}}_{\max}}{n}\right)D_f(x^k,x^*)+\left(\frac{4\widetilde{\mathcal{L}}\widetilde{\mathcal{L}}'_{\max}}{n}+\frac{2\widetilde{\mathcal{L}}_{\max}}{n}\right)\sigma^k,
\end{aligned}
$$

with the following choice of θ:

$$
\theta\coloneqq\frac{n\widetilde{\mathcal{L}}}{\widetilde{\mathcal{L}}_{\max}+2\widetilde{\mathcal{L}}\widetilde{\mathcal{L}}'_{\max}}\leq\frac{n}{2\widetilde{\mathcal{L}}'_{\max}},\quad\theta'\coloneqq\frac{2\theta}{n}\widetilde{\mathcal{L}}'_{\max}\leq 1.
$$

For the control vectors h_i^k and H^k, we deduce

$$
\begin{aligned}
&\mathbb{E}\left[\sigma^{k+1}\right] \\
&\quad\leq (1-\alpha)\frac{1}{n}\sum_{i=1}^{n}\|h_i^k-\nabla f_i(x^*)\|^2_{\mathbf{L}_i^\dagger}+2\alpha D_f(x^k,x^*) \\
&\qquad+(1-\beta)\theta\left\|H^k-\nabla f(x^*)\right\|^2_{\mathbf{L}^\dagger}+2\beta\theta\left(1+\frac{2\widetilde{\mathcal{L}}'_{\max}}{n}\right)D_f(x^k,x^*)+\frac{2\beta\theta\widetilde{\mathcal{L}}'_{\max}}{n^2}\sum_{i=1}^{n}\left\|h_i^k-\nabla f_i(x^*)\right\|^2_{\mathbf{L}_i^\dagger} \\
&\quad=\left(1-\alpha+\frac{2\beta\theta\widetilde{\mathcal{L}}'_{\max}}{n}\right)\frac{1}{n}\sum_{i=1}^{n}\|h_i^k-\nabla f_i(x^*)\|^2_{\mathbf{L}_i^\dagger}+(1-\beta)\theta\left\|H^k-\nabla f(x^*)\right\|^2_{\mathbf{L}^\dagger} \\
&\qquad+2\left(\alpha+\beta\theta\left(1+\frac{2\widetilde{\mathcal{L}}'_{\max}}{n}\right)\right)D_f(x^k,x^*) \\
&\quad\leq\max\left(1-\alpha+\frac{2\beta\theta\widetilde{\mathcal{L}}'_{\max}}{n},1-\beta\right)\sigma^k+2\left(\alpha+\beta\theta\left(1+\frac{2\widetilde{\mathcal{L}}'_{\max}}{n}\right)\right)D_f(x^k,x^*) \\
&\quad=\max\left(1-\alpha+\beta\theta',1-\beta\right)\sigma^k+2\left(\alpha+\beta\theta+\beta\theta'\right)D_f(x^k,x^*).
\end{aligned}
$$

Thus the constants from (Gorbunov et al., 2020a) are as follows

$$
\begin{aligned}
A&=L+2\widetilde{\mathcal{L}}+\frac{4\widetilde{\mathcal{L}}\widetilde{\mathcal{L}}'_{\max}}{n}+\frac{2\widetilde{\mathcal{L}}_{\max}}{n}\\
B&=\frac{4\widetilde{\mathcal{L}}\widetilde{\mathcal{L}}'_{\max}}{n}+\frac{2\widetilde{\mathcal{L}}_{\max}}{n}=\frac{2\widetilde{\mathcal{L}}}{\theta}\\
C&=\alpha+\beta\theta+\beta\theta'\\
\rho&=\min\left(\alpha-\beta\theta',\beta\right).
\end{aligned}
$$

Let $M=\frac{2B}{\rho}$, and note that $B\theta=2\widetilde{\mathcal{L}}$ and $B\theta'=\frac{4\widetilde{\mathcal{L}}\widetilde{\mathcal{L}}'_{\max}}{n}$. Then

$$
\begin{aligned}
A+CM&=A+2B\frac{\alpha+\beta\theta+\beta\theta'}{\min\left(\alpha-\beta\theta',\beta\right)}\\
&=\mathcal{O}\left(\frac{\alpha+\beta\theta+\beta\theta'}{\min\left(\alpha-\beta\theta',\beta\right)}\left(L+\widetilde{\mathcal{L}}+\frac{\widetilde{\mathcal{L}}\widetilde{\mathcal{L}}'_{\max}}{n}+\frac{\widetilde{\mathcal{L}}_{\max}}{n}\right)\right).\\
1+\frac{B}{M}-\rho&=1-\frac{\rho}{2}=1-\frac{1}{2}\min\left(\alpha-\beta\theta',\beta\right).
\end{aligned}
$$

□