MARINA Meets Matrix Stepsizes: Variance Reduced Distributed Non-Convex Optimization

Hanmin Li
AI Initiative
KAUST, Thuwal, Saudi Arabia
hanmin.li@kaust.edu.sa

Avetik Karagulyan
AI Initiative
KAUST, Thuwal, Saudi Arabia
avetik.karagulyan@kaust.edu.sa

Peter Richtárik
AI Initiative
KAUST, Thuwal, Saudi Arabia
peter.richtarik@kaust.edu.sa

Abstract

Matrix-stepsized gradient descent algorithms have been demonstrated to exhibit superior efficiency in non-convex optimization compared to their scalar counterparts. The `det-CGD` algorithm, as introduced by [LKR23], leverages matrix stepsizes to perform compressed gradient descent for non-convex objectives and matrix-smooth problems in a federated manner. The authors establish the algorithm's convergence to a neighborhood of the weighted stationarity point under a convex condition for the symmetric and positive-definite stepsize matrix. In this paper, we propose a variance-reduced version of the `det-CGD` algorithm, incorporating the `MARINA` method. Notably, we establish theoretically and empirically, that `det-MARINA` outperforms both `MARINA` and the distributed `det-CGD` algorithms

1 Introduction

1.1 Problem Setting

We are focusing on optimizing the finite sum non-convex objective, as defined below:

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

In this context, each function $f_i : \mathbb{R}^d \to \mathbb{R}$ is differentiable. It is assumed that f is lower bounded by $f^\star$, while each individual function f_i is lower bounded by $f_i^\star$. This type of objective function finds extensive application in various practical machine learning algorithms, which increase not only in terms of the data size but also in the model size and overall complexity as well. As a result, most neural network architectures result in highly non-convex empirical losses, which need to be minimized. In addition, it becomes computationally infeasible to train these models on one device, often excessively large, and one needs to redistribute them amongst different devices/clients. This redistribution results in a high communication overheads, which are often become the bottleneck in this framework.

In other words, we have the following setting. The data is partitioned into n clients, where the i-th client has access to the component function f_i and its derivatives. The clients are connected to each other through a central device, called the server. In this work, we are going to study iterative

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

gradient descent-based algorithms that operate as follows: The clients compute the local gradients in parallel. Then they compress these gradients to reduce the communication cost and send them to the server in parallel. The server then aggregates these vectors and broadcasts the iterate update back to the clients. This meta-algorithm is called federated learning. We refer the readers to [KMY+16, MMR+17, KMA+21] for a more thorough introduction to federated learning.

Background and related work. To solve the non-convex optimization problems, the most popular approaches use the SGD and its variants [JK+17, DDG+22]. For a more thorough background on these methods and an overview of prior work, we refer the readers to Appendix B and Appendix C, respectively.

In this work, we focus on an instance of the SGD algorithm called det-CGD. Proposed by [LKR23][1], the algorithm suggests using a uniform "upper bound" on the inverse Hessian matrix as its stepsize. Assuming matrix smoothness of the objective [SHR21], they replace the scalar stepsize with a positive definite matrix $\boldsymbol{D}$. The algorithm is given as follows:

$$x^{k+1} = x^k - \boldsymbol{D}\boldsymbol{S}^k \nabla f(x^k). \tag{det-CGD}$$

Here, $\boldsymbol{D}$ plays the role of the stepsize. In particular, if $\boldsymbol{D} = \gamma \boldsymbol{I}_d$, then we recover the standard scalar stepsize. The $\boldsymbol{S}^k$ are unbiased stochastic sketch matrices, that can be seen as sparsifiers of the gradient. The latter plays an important role in distributed setting.

The neighborhood of the distributed det-CGD1. The distributed version of det-CGD follows the standard federated learning paradigm [MMR+17]. The pseudocode of the method, as well as the convergence result of [LKR23], can be found in the appendix. Informally, their convergence result can be written as

$$\min_{k=1,\ldots,K} \mathbb{E}[\left\|\nabla f(x^k)\right\|_{\boldsymbol{D}}^2] \leq \mathcal{O}\left(\frac{(1+\alpha)^K}{K}\right) + \mathcal{O}(\alpha), \tag{2}$$

where $\alpha > 0$ is a constant that can be controlled. The crucial insight from this result is that the error bound doesn't diminish as the number of iterations increases. In fact, by controlling α and considering a large K, it's impossible to make the second term smaller than ε. This implies that the algorithm converges to a certain neighborhood surrounding the (local) optimum. This phenomenon is a common occurrence in SGD and is primarily attributable to the variance associated with the stochastic gradient estimator. In the case of det-CGD the stochasticity comes from the sketch $\boldsymbol{S}^k$.

Contributions. In this paper, we introduce a federated learning algorithm named det-MARINA. This algorithm extends a recent method called det-CGD [LKR23], which aims to solve (1) using matrix stepsized gradient descent. Under the matrix smoothness assumption [SHR21], they demonstrate that the matrix stepsized version of the Distributed Compressed Gradient Descent [KFJ18] algorithm enhances communication complexity compared to its scalar counterpart. However, in their analysis, [LKR23] show stationarity only within a certain neighborhood due to stochastic compressors. Our algorithm addresses this issue by incorporating a variance reduction scheme called MARINA [GBLR21], which is an optimal federated learning algorithm in the non-convex setting. We establish theoretically and empirically, that det-MARINA outperforms both MARINA and the distributed det-CGD algorithms in terms of iteration and communication complexities. In addition, we describe specific matrix stepsize choices, for which algorithm beats the scalar MARINA and distributed det-CGD both in theory and in practice.

2 Mathematical Framework

We refer the readers to Appendix A.1 for mathematical notations and basic facts.

2.1 Assumptions and Conditions

In this section we present the assumptions we needed in order to analyze det-MARINA.

[1] In the original paper, the algorithm is referred to as det-CGD, as there is a variant of the same algorithm named det-CGD2. Since we are going to use only the first one and our framework is applicable to both, we will remove the number in the end for the sake of brevity.

Assumption 1. (Lower Bound) *There exists $f^\star \in \mathbb{R}$ such that, $f(x) \geq f^\star$ for all $x \in \mathbb{R}^d$.*

This is a standard assumption in optimization, as otherwise the problem of minimizing the objective would not be correct mathematically. The same assumption is used in MARINA. We then need a matrix version of L-smoothness in order to proceed. Previously, [SHR21], [WSR22] used $\boldsymbol{L}$-matrix smoothness in the (strongly) convex setting to analyze some variants of the DCGD. [LKR23] provided the analysis of sketched gradient descent under this assumption in the non-convex case. The assumption is formulated as follows,

Assumption 2. ($\boldsymbol{L}_i$-matrix smoothness) *Assume that each function f_i is $\boldsymbol{L}_i$-smooth for all $i \in [n] = \{1, 2, 3, ..., n\}$. That is for each function f_i, the following inequality holds:*

$$f_i(x) \leq f_i(y) + \langle \nabla f_i(y), x - y \rangle + \frac{1}{2} \|x - y\|^2_{\boldsymbol{L}_i}. \tag{3}$$

However, we do not want to use Assumption 2 in the analysis, as in the analysis of det-MARINA one needs to upper bound the squared difference of gradients by a multiple of squared difference of iterate. The latter is not implied from Assumption 2 for non-convex functions, as opposed to the convex ones. Instead, we introduce the matrix version of the L-Lipschitz continuous gradient assumption used in the analysis for MARINA of [GBLR21]. Note that the "smoothness" the authors are referring to is indeed the Lipschitz continuous gradient assumption, instead of the standard smoothness assumption [Nes03].

Definition 1. ($\boldsymbol{L}$-Lipschitz Gradient) *Assume that $f : \mathbb{R}^d \to \mathbb{R}$ is a continuously differentiable function and matrix $\boldsymbol{L} \in \mathbb{S}^d_{++}$. We say the gradient of f is $\boldsymbol{L}$-Lipschitz if*

$$\|\nabla f(x) - \nabla f(y)\|_{\boldsymbol{L}^{-1}} \leq \|x - y\|_{\boldsymbol{L}}, \quad \forall x, y \in \mathbb{R}^d. \tag{4}$$

This condition can be interpreted as follows. The gradient of f naturally belongs to the dual space of $\mathbb{R}^d$, as it is defined as a linear functional on $\mathbb{R}^d$. In the scalar case, ℓ_2-norm is self-dual, thus (4) reduces to the standard Lipschitz continuity of the gradient. However, with the matrix smoothness assumption, we are using the $\boldsymbol{L}$-norm for the iterates, which naturally induces the $\boldsymbol{L}^{-1}$-matrix norm for the gradients in the dual space. This insight, which is originally presented by [NY83], plays a key role in our analysis.

The following proposition provides us with a method to verify (4).

Proposition 1. *Given twice continuously differentiable function $f : \mathbb{R}^d \mapsto \mathbb{R}$ with bounded Hessian,*

$$\nabla^2 f(x) \preceq \boldsymbol{L}, \tag{5}$$

where $\boldsymbol{L} \in \mathbb{S}^d_{++}$ and the generalized inequality holds for any $x \in \mathbb{R}^d$. Then f satisfies (4) with the matrix $\boldsymbol{L}$.

Despite, being equivalent in the convex setting, (4) is slightly stronger compared to Assumption 2 in the non-convex case. See Appendix D for the properties of matrix smoothness. However, in practical terms, verifying Proposition 1 serves as the pipeline for confirming both conditions. Finally, we check that (4) is indeed an extension of the standard Lipschitz gradient assumption, as illustrated by the following remark.

Remark 1. *If we let $\boldsymbol{L} = L\boldsymbol{I}_d$, then (4) reduces to the standard L-Lipschitz continuous gradient assumption.*

In the following, we will assume that (4) is satisfied for component functions f_i.

Assumption 3. *Each function f_i is $\boldsymbol{L}_i$-gradient Lipschitz, while f is $\boldsymbol{L}$-gradient Lipschitz.*

In fact, the second half of the assumption is a consequence of the first one. Below, we formalize this claim.

Proposition 2. *If f_i is $\boldsymbol{L}_i$-gradient Lipschitz for every $i = 1, \ldots, n$, then function f has $\boldsymbol{L}$-Lipschitz gradient with $\boldsymbol{L} \in \mathbb{S}^d_{++}$ satisfying*

$$\frac{1}{n} \sum_{i=1}^{n} \lambda_{\max}\left(\boldsymbol{L}^{-1}\right) \cdot \lambda_{\max}\left(\boldsymbol{L}_i\right) \cdot \lambda_{\max}\left(\boldsymbol{L}_i \boldsymbol{L}^{-1}\right) = 1. \tag{6}$$

Nevertheless, the matrix $\boldsymbol{L}$ found according to Proposition 2 is only an estimate. In principle, there might exist a better $\boldsymbol{L}_f \preceq \boldsymbol{L}$ such that f has $\boldsymbol{L}_f$-Lipschitz gradient.

Remark 2. *In the scalar case, where $\boldsymbol{L} = L\boldsymbol{I}_d$, $\boldsymbol{L}_i = L_i\boldsymbol{I}_d$, the relation becomes*

$$L^2 = \frac{1}{n}\sum_{i=1}^{n} L_i^2.$$

This corresponds to the statement in Assumption 1.2 in [GBLR21].

3 The Main Result

In this section, we present our algorithm `det-MARINA` with the main convergence result. In addition, we compute both iteration and communication complexities and show that they are better than the ones of the `MARINA` algorithm, which serves as the prototype of our method. Along the iterations of the algorithms, we are constructing a sequence of vectors g^k which are stochastic estimators of $\nabla f(x^k)$. At each iteration, the server samples a Bernoulli random variable (coin flip) c_k and broadcasts it in parallel to the clients, along with the current gradient estimate g^k. Each client, then, does a `det-CGD`-type update with the stepsize $\boldsymbol{D}$ and a gradient estimate g^k. The next gradient estimate g^{k+1} is then computed. With a low probability, that is when $c_k = 1$, we take the g^{k+1} to be the full gradient $\nabla f(x^{k+1})$. Otherwise, we update it using the compressed gradient differences at each client. Below, is the pseudocode of the algorithm.

Algorithm 1 `det-MARINA`

Input: starting point x^0, stepsize matrix $\boldsymbol{D}$, probability $p \in (0,1]$, number of iterations K
Initialize $g^0 = \nabla f(x^0)$
for $k = 0, 1, \ldots, K-1$ **do**
  Sample $c_k \sim \text{Be}(p)$
  Broadcast g^k to all workers
  for $i = 1, 2, \ldots$ in parallel **do**
    $x^{k+1} = x^k - \boldsymbol{D}\cdot g^k$
    Set $g_i^{k+1} = \begin{cases} \nabla f_i(x^{k+1}) & \text{if } c_k = 1 \\ g^k + \boldsymbol{S}_i^k\left(\nabla f_i(x^{k+1}) - \nabla f_i(x^k)\right) & \text{if } c_k = 0 \end{cases}$
  end for
  $g^{k+1} = \frac{1}{n}\sum_{i=1}^n g_i^{k+1}$
end for
Return: $\tilde{x}^K$ chosen uniformly at random from $\{x^k\}_{k=0}^{K-1}$

In the following theorem, we formulate the main result of this paper, which guarantees the convergence of Algorithm 1 under the abovementioned assumptions.

Theorem 1. *Assume that Assumptions 1 and 3 hold, and the following condition on stepsize matrix $\boldsymbol{D} \in \mathbb{S}^d_{++}$ holds,*

$$\boldsymbol{D}^{-1} \succeq \left(\frac{(1-p)\cdot R(\boldsymbol{D},\mathcal{S})}{np} + 1\right)\boldsymbol{L}, \tag{7}$$

where

$$R(\boldsymbol{D},\mathcal{S}) := \frac{1}{n}\sum_{i=1}^{n} \lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{D}\boldsymbol{S}_i^k\right] - \boldsymbol{D}\right)\cdot\lambda_{\max}\left(\boldsymbol{L}_i\right)\cdot\lambda_{\max}\left(\boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{L}_i\boldsymbol{L}^{-\frac{1}{2}}\right). \tag{8}$$

Then, after K-iterations of `det-MARINA`*, we have*

$$\mathbb{E}\left[\left\|\nabla f(\tilde{x}^K)\right\|^2_{\frac{\boldsymbol{D}}{\det(\boldsymbol{D})^{1/d}}}\right] \leq \frac{2\left(f(x^0) - f^\star\right)}{\det(\boldsymbol{D})^{1/d}\cdot K}. \tag{9}$$

Here, $\tilde{x}^K$ is chosen uniformly randomly from the first K iterates of the algorithm.

Below we state several remarks regarding the interpretation of theorem.

Remark 3. *We notice that the the right-hand side of the algorithm vanishes with the number of iterations, thus solving the issue of the distributed* `det-CGD`*. Therefore,* `det-MARINA` *is indeed the variance reduced version of* `det-CGD` *in the distributed setting and has better convergence guarantees.*

Remark 4. *Theorem 1 implies the following iteration complexity for the algorithm. In order to get an ε^2 stationarity error, the algorithm requires K iterations, with*

$$K \geq \frac{2(f(x^0) - f^\star)}{\det(\boldsymbol{D})^{1/d} \cdot \varepsilon^2}.$$

Remark 5. *In the case where no compression is applied, that is we have $\boldsymbol{S}_i^k = \boldsymbol{I}_d$, the condition (7) reduces to*

$$\boldsymbol{D} \preceq \boldsymbol{L}^{-1}. \tag{10}$$

The latter is due to $\mathbb{E}\left[\boldsymbol{S}_i^k \boldsymbol{D} \boldsymbol{S}_i^k\right] = \boldsymbol{D}$, which results in $R(\boldsymbol{D}, \mathcal{S}) = 0$. This is expected, since in the deterministic case `det-MARINA` *reduces to* `GD` *with matrix stepsize.*

The convergence condition and rate of matrix stepsize `GD` can be found in [LKR23]. Below we do a sanity check to verify that the convergence condition for scalar `MARINA` can be obtained.

Remark 6. *Let us consider the scalar case. That is*

$$\boldsymbol{L}_i = L_i \boldsymbol{I}_d, \quad \boldsymbol{L} = L\boldsymbol{I}_d, \quad \boldsymbol{D} = \gamma \boldsymbol{I}_d \quad \textit{and} \quad \omega = \lambda_{\max}\left(\mathbb{E}\left[\left(\boldsymbol{S}_i^k\right)^\top \boldsymbol{S}_i^k\right]\right) - 1. \tag{11}$$

Then, the condition (7) reduces to

$$\frac{\gamma(1-p)\omega L^2}{np} - \frac{1}{\gamma} + L \leq 0. \tag{12}$$

One can check that the below bound implies (12)

$$\gamma \leq \frac{1}{L\left(1 + \sqrt{\frac{(1-p)\omega}{pn}}\right)}.$$

The latter coincides with the stepsize condition of the convergence result of scalar `MARINA`. Now let us look at the right-hand side of (9). We notice that it decreases in terms of the determinant of the stepsize matrix. Therefore, one needs to solve the following optimization problem to find the optimal stepsize:

$$\begin{array}{ll} \text{minimize} & \log\det(\boldsymbol{D}^{-1}) \\ \text{subject to} & \boldsymbol{D} \text{ satisfying (7).} \end{array}$$

The solution of this constrained minimization problem on $\mathbb{S}^d_{++}$ is not explicit. In theory, one may show that the constraint (7) is convex and attempt to solve the problem numerically. However, as stressed by [LKR23], the similar stepsize condition for `det-CGD` is not easily computed using solvers like CVXPY [DB16]. Instead, we may relax the problem to certain linear subspaces of $\mathbb{S}^d_{++}$. In particular, we fix a matrix $\boldsymbol{W} \in \mathbb{S}^d_{++}$, and define $\boldsymbol{D} := \gamma \boldsymbol{W}$. Then, the condition on the matrix $\boldsymbol{D}$ becomes a condition for the scalar γ, which is given in the following corollary.

Corollary 1. *Let $\boldsymbol{W} \in \mathbb{S}^d_{++}$, defining $\boldsymbol{D} := \gamma \cdot \boldsymbol{W}$, where $\gamma \in \mathbb{R}_+$. then the condition in (7) reduces to the following condition on γ*

$$\gamma \leq \frac{2\lambda_{\boldsymbol{W}}}{1 + \sqrt{1 + 4\alpha\beta \cdot \Lambda_{\boldsymbol{W},\mathcal{S}}\lambda_{\boldsymbol{W}}}}, \tag{13}$$

where $\Lambda_{\boldsymbol{W},\mathcal{S}} = \lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k \boldsymbol{W} \boldsymbol{S}_i^k\right] - \boldsymbol{W}\right)$, $\lambda_{\boldsymbol{W}} = \lambda_{\max}^{-1}\left(\boldsymbol{W}^{\frac{1}{2}} \boldsymbol{L} \boldsymbol{W}^{\frac{1}{2}}\right)$ and

$$\alpha = \frac{1-p}{np}; \quad \beta = \frac{1}{n}\sum_{i=1}^n \lambda_{\max}\left(\boldsymbol{L}_i\right) \cdot \lambda_{\max}\left(\boldsymbol{L}^{-1}\boldsymbol{L}_i\right). \tag{14}$$

This means that for every fixed $\boldsymbol{W}$, we can find the optimal scaling γ. In the next section, we will use this corollary to compute the communication complexity of our algorithm and to compare it with `MARINA`.

4 Comparison of Complexities

The following corollary states the iteration complexity for `det-MARINA` with $\boldsymbol{W} = \boldsymbol{L}^{-1}$.

Corollary 2. *If we take $\boldsymbol{W} = \boldsymbol{L}^{-1}$, then the condition (13) on γ is given by*

$$\gamma \leq \frac{2}{1 + \sqrt{1 + 4\alpha\beta \cdot \Lambda_{\boldsymbol{L}^{-1},\mathcal{S}}}}. \tag{15}$$

In order to satisfy, ε-stationarity, that is $\mathbb{E}\left[\left\|\nabla f(\tilde{x}^K)\right\|^2_{\boldsymbol{D}/\det(\boldsymbol{D})^{1/d}}\right] \leq \varepsilon^2$, we require

$$K \geq K_0 = \mathcal{O}\left(\frac{\Delta_0 \cdot \det(\boldsymbol{L})^{\frac{1}{d}}}{\varepsilon^2} \cdot \left(1 + \sqrt{1 + 4\alpha\beta \cdot \Lambda_{\boldsymbol{L}^{-1},\mathcal{S}}}\right)\right), \tag{16}$$

where $\Delta_0 = f(x^0) - f(x^\star)$. Moreover, this iteration complexity is always better than the one of `MARINA`.

The proof can be found in the Appendix. In fact, we can show that in cases where we fix $\boldsymbol{W} = \boldsymbol{I}_d$ and $\boldsymbol{W} = \operatorname{diag}^{-1}(\boldsymbol{L})$, the same conclusion also holds, relevant details can be found in Appendix E.2. This essentially means that `det-MARINA` can always have a "larger" stepsize compared to `MARINA`, which leads to a better iteration complexity. In addition, because we are using the same compressor for those two algorithms, the communication complexity of `det-MARINA` is also provably better than that of `MARINA`.

We also give an analysis of the communication complexity of our algorithm as our main concern here is the communication complexity. We first give the following definition on the expected density, which is used to analyze the communication complexity in [GBLR21]. The original definition is given for any unbiased compressors. However, we are focusing on sketches in this paper, so we only restrict the definition to sketches.

Definition 2. *For a given sketch matrix $\boldsymbol{S} \in \mathbb{S}^d_+$, the expected density is defined as*

$$\zeta_{\boldsymbol{S}} = \sup_{x \in \mathbb{R}^d} \mathbb{E}\left[\left\|\boldsymbol{S}x\right\|_0\right], \tag{17}$$

where $\|x\|_0$ denotes the number of non-zero components of $x \in \mathbb{R}^d$.

We can easily obtain the expected density for some commonly seen sketches, for example for rand-τ sketches, we have $\zeta_{\texttt{rand-}\tau} = \tau$. The latter means, that in average the clients communicate τ coordinates at each iteration. Below, we state the communication complexity of `det-MARINA` with $\boldsymbol{D}^*_{\boldsymbol{L}^{-1}}$ and the rand-τ compressor.

Corollary 3. *Assume that we are using sketch $\boldsymbol{S} \sim \mathcal{S}$ with expected density $\zeta_{\mathcal{S}}$, suppose we are running* `det-MARINA` *with probability p and we use the optimal stepsize matrix with respect to $\boldsymbol{W} = \boldsymbol{L}^{-1}$, then the overall communication complexity here is given by $\mathcal{O}\left((Kp+1)d + (1-p)K\zeta_{\mathcal{S}}\right)$. Specifically, if we pick $p = \zeta_{\mathcal{S}}/d$, then the communication complexity is given by*

$$\mathcal{O}\left(d + \frac{\Delta_0 \cdot \det(\boldsymbol{L})^{\frac{1}{d}}}{\varepsilon^2} \cdot \left(\zeta_{\mathcal{S}} + \sqrt{\frac{\beta \cdot \Lambda_{\boldsymbol{L}^{-1},\mathcal{S}}}{n} \cdot \zeta_{\mathcal{S}}(d - \zeta_{\mathcal{S}})}\right)\right). \tag{18}$$

Notice that in case where no compression is applied, the communication complexity (resp. iteration complexity) reduces to $\mathcal{O}\left(d\Delta_0 \cdot \det(\boldsymbol{L})^{\frac{1}{d}}/\varepsilon^2\right)$ (resp. $\mathcal{O}\left(\Delta_0 \cdot \det(\boldsymbol{L})^{\frac{1}{d}}/\varepsilon^2\right)$), this coincides with the rate of matrix stepsize GD (see [LKR23]). This implies that the dependence on ε is not possible to be improved further since `GD` is optimal in first order methods [CDHS20]. Finally we conduct numerical experiments to support our findings, which are deferred to the Appendix I.

5 Future Work

i) In this paper, we have only considered (linear) sketches as the compression operator. However, there exists a variety of compressors which are useful in practice that do not fall into this category. Extending `det-CGD` and `det-MARINA` for general unbiased compressors is a promising future work

direction. ii) Our motivation for using a matrix stepsize is partially inspired by second-order methods, where matrix stepsize $\boldsymbol{D}$ roughly estimates the inverse of the Hessian. Additionally, given recent successes with adaptive stepsizes (e.g., [LVLLJ21, OLJL22, SGU23]), designing an adaptive matrix stepsize tailored to our case could be viable. iii) Finally, recent advances in federated learning [PD20, GTR22] have shown that server-to-client compression is important. Extending our results for the bidirectional federated learning is worth human attention.

References

[ABK07] Mehiddin Al-Baali and H Khalfan. An overview of some practical quasi-newton methods for unconstrained optimization. *Sultan Qaboos University Journal for Science [SQUJS]*, 12(2):199–209, 2007.

[ABSM14] Mehiddin Al-Baali, Emilio Spedicato, and Francesca Maggioni. Broyden's quasi-Newton methods for a nonlinear system of equations and unconstrained optimization: a review and open problems. *Optimization Methods and Software*, 29(5):937–954, 2014.

[ADA21] Foivos Alimisis, Peter Davies, and Dan Alistarh. Communication-efficient distributed optimization with quantized preconditioners. In *International Conference on Machine Learning*, pages 196–206. PMLR, 2021.

[AGL+17] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: Communication-efficient SGD via gradient quantization and encoding. *Advances in neural information processing systems*, 30, 2017.

[AZ17] Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1200–1205, 2017.

[B+15] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[Bha09] Rajendra Bhatia. *Positive definite matrices*. Princeton university press, 2009.

[Bro65] Charles G Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of computation*, 19(92):577–593, 1965.

[CCS10] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982, 2010.

[CDHS20] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, 184(1-2):71–120, 2020.

[CKK+19] Sélim Chraibi, Ahmed Khaled, Dmitry Kovalev, Peter Richtárik, Adil Salim, and Martin Takáč. Distributed fixed point methods with compressed iterates. *arXiv preprint arXiv:1912.09925*, 2019.

[CL11] 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.

[CR19] Rixon Crane and Fred Roosta. Dingo: Distributed newton-type method for gradient-norm optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

[DB16] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research*, 17(1):2909–2913, 2016.

[DBLJ14] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

[DDG+22] Marina Danilova, Pavel Dvurechensky, Alexander Gasnikov, Eduard Gorbunov, Sergey Guminov, Dmitry Kamzolov, and Innokentiy Shibaev. Recent theoretical advances in non-convex optimization. In *High-Dimensional Optimization and Probability: With a View Towards Data Science*, pages 79–163. Springer, 2022.

[DM77] John E Dennis, Jr and Jorge J Moré. Quasi-Newton methods, motivation and theory. *SIAM review*, 19(1):46–89, 1977.

[DOG+19] Darina Dvinskikh, Aleksandr Ogaltsov, Alexander Gasnikov, Pavel Dvurechensky, Alexander Tyurin, and Vladimir Spokoiny. Adaptive gradient descent for convex and non-convex stochastic optimization. *arXiv preprint arXiv:1911.08380*, 2019.

[DTN+20] Canh T Dinh, Nguyen H Tran, Tuan Dung Nguyen, Wei Bao, Albert Y Zomaya, and Bing B Zhou. Federated learning with proximal stochastic variance reduced gradient algorithms. In *Proceedings of the 49th International Conference on Parallel Processing*, pages 1–11, 2020.

[GBLR21] Eduard Gorbunov, Konstantin P Burlachenko, Zhize Li, and Peter Richtárik. Marina: Faster non-convex distributed learning with compression. In *International Conference on Machine Learning*, pages 3788–3798. PMLR, 2021.

[GLQ+19] Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. Sgd: General analysis and improved rates. In *International Conference on Machine Learning*, pages 5200–5209. PMLR, 2019.

[GNDG19] SV Guminov, Yu E Nesterov, PE Dvurechensky, and AV Gasnikov. Accelerated primal-dual gradient descent with linesearch for convex, nonconvex, and nonsmooth optimization problems. In *Doklady Mathematics*, volume 99, pages 125–128. Springer, 2019.

[GSBR20] 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.

[GT74] William B Gragg and Richard A Tapia. Optimal error bounds for the Newton–Kantorovich theorem. *SIAM Journal on Numerical Analysis*, 11(1):10–13, 1974.

[GTR22] Kaja Gruntkowska, Alexander Tyurin, and Peter Richtárik. EF21-P and Friends: Improved Theoretical Communication Complexity for Distributed Optimization with Bidirectional Compression, 2022.

[HHH+22] Samuel Horváth, Chen-Yu Ho, Ludovit Horvath, Atal Narayan Sahu, Marco Canini, and Peter Richtárik. Natural compression for distributed deep learning. In *Mathematical and Scientific Machine Learning*, pages 129–141. PMLR, 2022.

[HKM+23] Samuel Horváth, Dmitry Kovalev, Konstantin Mishchenko, Peter Richtárik, and Sebastian Stich. Stochastic distributed learning with gradient quantization and double-variance reduction. *Optimization Methods and Software*, 38(1):91–106, 2023.

[HR20] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020.

[IQR21] Rustem Islamov, Xun Qian, and Peter Richtárik. Distributed second order methods with fast rates and compressed communication. In *International conference on machine learning*, pages 4617–4628. PMLR, 2021.

[JK+17] Prateek Jain, Purushottam Kar, et al. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142–363, 2017.

[JRSPS16] Sashank J Reddi, Suvrit Sra, Barnabas Poczos, and Alexander J Smola. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. *Advances in neural information processing systems*, 29, 2016.

[JZ13] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

[KFJ18] Sarit Khirirat, Hamid Reza Feyzmahdavian, and Mikael Johansson. Distributed learning with compressed gradients. *arXiv preprint arXiv:1806.06573*, 2018.

[KHR20] 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 *Algorithmic Learning Theory*, pages 451–467. PMLR, 2020.

[KMA+21] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[KMY+16] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[KR20] Ahmed Khaled and Peter Richtárik. Better theory for sgd in the nonconvex world. *arXiv preprint arXiv:2002.03329*, 2020.

[LBZR21] Zhize Li, Hongyan Bao, Xiangliang Zhang, and Peter Richtárik. PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *International conference on machine learning*, pages 6286–6295. PMLR, 2021.

[LKQR20] Zhize Li, Dmitry Kovalev, Xun Qian, and Peter Richtárik. Acceleration for compressed gradient descent in distributed and federated optimization. *arXiv preprint arXiv:2002.11364*, 2020.

[LKR23] Hanmin Li, Avetik Karagulyan, and Peter Richtárik. Det-cgd: Compressed gradient descent with matrix stepsizes for non-convex optimization. *arXiv preprint arXiv:2305.12568*, 2023.

[LVLLJ21] Nicolas Loizou, Sharan Vaswani, Issam Hadj Laradji, and Simon Lacoste-Julien. Stochastic polyak step-size for sgd: An adaptive learning rate for fast convergence. In *International Conference on Artificial Intelligence and Statistics*, pages 1306–1314. PMLR, 2021.

[Mai15] Julien Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

[MGTR19] Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*, 2019.

[Mie80] George J Miel. Majorizing sequences and error bounds for iterative methods. *Mathematics of Computation*, 34(149):185–202, 1980.

[MMR+17] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[MMSR22] Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich, and Peter Richtarik. ProxSkip: Yes! Local gradient steps provably lead to communication acceleration! Finally! In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15750–15769. PMLR, 17–23 Jul 2022.

[MSR22] Artavazd Maranjyan, Mher Safaryan, and Peter Richtárik. GradSkip: Communication-Accelerated Local Gradient Methods with Better Computational Complexity. *arXiv preprint arXiv:2210.16402*, 2022.

[Nes03] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

[Nes12] Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[NJLS09] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

[NY83] Arkadi Semenovič Nemirovski and David Borisovich Yudin. Problem complexity and method efficiency in optimization. *Wiley-Interscience, ISSN 0277-2698*, 1983.

[OLJL22] Antonio Orvieto, Simon Lacoste-Julien, and Nicolas Loizou. Dynamics of sgd with stochastic polyak stepsizes: Truly adaptive variants and convergence to exact solution. *Advances in Neural Information Processing Systems*, 35:26943–26954, 2022.

[PD20] Constantin Philippenko and Aymeric Dieuleveut. Bidirectional compression in heterogeneous settings for distributed or federated learning with partial participation: tight convergence guarantees. *arXiv preprint arXiv:2006.14591*, 2020.

[PWLC22] Jie Peng, Zhaoxian Wu, Qing Ling, and Tianyi Chen. Byzantine-robust variance-reduced federated learning over distributed non-iid data. *Information Sciences*, 616:367–391, 2022.

[QQR21] Xun Qian, Zheng Qu, and Peter Richtárik. L-svrg and l-katyusha with arbitrary sampling. *The Journal of Machine Learning Research*, 22(1):4991–5039, 2021.

[RT14] Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

[SGU23] Fabian Schaipp, Robert M Gower, and Michael Ulbrich. A stochastic proximal polyak step size. *arXiv preprint arXiv:2301.04935*, 2023.

[SHR21] Mher Safaryan, Filip Hanzely, and Peter Richtárik. Smoothness matrices beat smoothness constants: Better communication compression techniques for distributed optimization. *Advances in Neural Information Processing Systems*, 34:25688–25702, 2021.

[SIQR22] Mher Safaryan, Rustem Islamov, Xun Qian, and Peter Richtarik. Fednl: Making newton-type methods applicable to federated learning. In *International Conference on Machine Learning*, pages 18959–19010. PMLR, 2022.

[SLRB17] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162:83–112, 2017.

[SR22] Egor Shulgin and Peter Richtárik. Shifted compression framework: Generalizations and improvements. In *Uncertainty in Artificial Intelligence*, pages 1813–1823. PMLR, 2022.

[SSR22] Mher Safaryan, Egor Shulgin, and Peter Richtárik. Uncertainty principle for communication compression in distributed and federated learning and the search for an optimal compressor. *Information and Inference: A Journal of the IMA*, 11(2):557–580, 2022.

[Tib96] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

[TR22] Alexander Tyurin and Peter Richtárik. DASHA: Distributed nonconvex optimization with communication compression, optimal oracle complexity, and no client synchronization. *arXiv preprint arXiv:2202.01268*, 2022.

[Wri15] Stephen J Wright. Coordinate descent algorithms. *Mathematical programming*, 151(1):3–34, 2015.

[WRXM18] Shusen Wang, Fred Roosta, Peng Xu, and Michael W Mahoney. Giant: Globally improved approximate newton method for distributed optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[WSR22] Bokun Wang, Mher Safaryan, and Peter Richtárik. Theoretically better and numerically faster distributed optimization with smoothness-aware quantization techniques. *Advances in Neural Information Processing Systems*, 35:9841–9852, 2022.

[Yam87] Tetsuro Yamamoto. A convergence theorem for newton-like methods in banach spaces. *Numerische Mathematik*, 51:545–557, 1987.

[ZKV+20] Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33:15383–15393, 2020.

[ZYB20] Jiaqi Zhang, Keyou You, and Tamer Başar. Achieving globally superlinear convergence for distributed optimization with adaptive newton method. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 2329–2334. IEEE, 2020.

Contents

A Notations and Basic Facts

A.1 Notations

The standard Euclidean norm on $\mathbb{R}^d$ is defined as $\|\cdot\|$. We use $\mathbb{S}^d_{++}$ (resp. $\mathbb{S}^d_+$) to denote the positive definite (resp. semi-definite) cone of dimension d. $\mathbb{S}^d$ is used to denote all symmetric matrices of dimension d. We use the notation $\boldsymbol{I}_d$ to denote the identity matrix of size $d \times d$, and $\boldsymbol{O}_d$ to denote the zero matrix of size $d \times d$. Given $\boldsymbol{Q} \in \mathbb{S}^d_{++}$ and $x \in \mathbb{R}^d$, $\|x\|_{\boldsymbol{Q}} := \sqrt{x^\top \boldsymbol{Q} x} = \sqrt{\langle x, \boldsymbol{Q}x\rangle}$, where $\langle \cdot, \cdot \rangle$ is the standard Euclidean inner product on $\mathbb{R}^d$. For a matrix $\boldsymbol{A} \in \mathbb{S}^d$, we use $\lambda_{\max}(\boldsymbol{A})$ (resp. $\lambda_{\min}(\boldsymbol{A})$) to denote the largest (resp. smallest) eigenvalue of the matrix $\boldsymbol{A}$. For a function $f : \mathbb{R}^d \mapsto \mathbb{R}$, its gradient and its Hessian at a point $x \in \mathbb{R}^d$ are respectively denoted as $\nabla f(x)$ and $\nabla^2 f(x)$. For the sketch matrices $\boldsymbol{S}^k_i$ used in the algorithm, we use the superscript k to denote the iteration and subscript i to denote the client, the matrix $\boldsymbol{S}^k_i$ is thus sampled for client i in the k-th iteration from sine distribution $\mathcal{S}$. For any matrix $\boldsymbol{A} \in \mathbb{S}^d$, we use the notation $\operatorname{diag}(\boldsymbol{A}) \in \mathbb{S}^d$ to denote the diagonal of matrix $\boldsymbol{A}$.

A.2 Basic Facts

In this section, we present some basic facts along with their proofs that will be used later in the analysis.

Fact 1. *For two matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{S}^d_+$, denote the i-th largest eigenvalues of $\boldsymbol{A}, \boldsymbol{B}$ as $\lambda_i(\boldsymbol{A}), \lambda_i(\boldsymbol{B})$, if $\boldsymbol{A} \succeq \boldsymbol{B}$, then the following holds*

$$\lambda_i(\boldsymbol{A}) \geq \lambda_i(\boldsymbol{B}). \tag{19}$$

Proof. According to the Courant-Fischer theorem, we write

$$\lambda_i(\boldsymbol{B}) = \max_{S:\dim S = i} \min_{x \in S \setminus \{0\}} \frac{x^\top \boldsymbol{B} x}{x^\top x}.$$

Let $S^i_{\max}$ be a subspace of dimension i where the maximum is attained, we then have

$$\begin{aligned}
\lambda_i(\boldsymbol{B}) &= \min_{x \in S^i_{\max} \setminus \{0\}} \frac{x^\top \boldsymbol{B} x}{x^\top x} \\
&\leq \min_{x \in S^i_{\max} \setminus \{0\}} \frac{x^\top \boldsymbol{A} x}{x^\top x} \\
&\leq \max_{S:\dim S = i} \min_{x \in S \setminus \{0\}} \frac{x^\top \boldsymbol{A} x}{x^\top x} = \lambda_i(\boldsymbol{A}).
\end{aligned}$$

This completes the proof. □

The following is a generalization of the bias-variance decomposition for the matrix norm.

Fact 2. (Variance Decomposition) *Given a matrix $\boldsymbol{M} \in \mathbb{S}^d_{++}$, any vector $c \in \mathbb{R}^d$, and a random vector $x \in \mathbb{R}^d$ such that $\mathbb{E}\left[\|x\|\right] \leq +\infty$, the following bound holds*

$$\mathbb{E}\left[\|x - \mathbb{E}\left[x\right]\|^2_{\boldsymbol{M}}\right] = \mathbb{E}\left[\|x - c\|^2_{\boldsymbol{M}}\right] - \|\mathbb{E}\left[x\right] - c\|^2_{\boldsymbol{M}}. \tag{20}$$

Proof. We have

$$\begin{aligned}
&\mathbb{E}\left[\|x - c\|^2_{\boldsymbol{M}}\right] - \|\mathbb{E}\left[x\right] - c\|^2_{\boldsymbol{M}} \\
&\qquad = \mathbb{E}\left[x^\top \boldsymbol{M} x\right] - 2\mathbb{E}\left[x\right]^\top \boldsymbol{M} c + c^\top \boldsymbol{M} c - \mathbb{E}\left[x\right]^\top \boldsymbol{M}\mathbb{E}\left[x\right] + 2\mathbb{E}\left[x\right]^\top \boldsymbol{M} c - c^\top \boldsymbol{M} c \\
&\qquad = \mathbb{E}\left[x^\top \boldsymbol{M} x\right] - \mathbb{E}\left[x\right]^\top \boldsymbol{M}\mathbb{E}\left[x\right] \\
&\qquad = \mathbb{E}\left[x^\top \boldsymbol{M} x\right] - 2 \cdot \mathbb{E}\left[x\right]^\top \boldsymbol{M}\mathbb{E}\left[x\right] + \mathbb{E}\left[x\right]^\top \boldsymbol{M}\mathbb{E}\left[x\right] \\
&\qquad = \mathbb{E}\left[\|x - \mathbb{E}\left[x\right]\|^2_{\boldsymbol{M}}\right].
\end{aligned}$$

This completes the proof. □

Fact 3. *The map $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{X}) \mapsto \boldsymbol{A} - \boldsymbol{X}\boldsymbol{B}^{-1}\boldsymbol{X}$ is jointly concave on $\mathbb{S}^d_+ \times \mathbb{S}^d_{++} \times \mathbb{S}^d$. It is also monotone increasing in variables $\boldsymbol{A}$ and $\boldsymbol{B}$.*

We refer the reader to [Bha09, Corollary 1.5.3] for the details and the proof. The following is a result of Fact 1 and Fact 3.

Fact 4. *Suppose $\boldsymbol{L}_i \in \mathbb{S}^d_{++}$, for $i = 1, \ldots, n$. Then, for every matrix $\boldsymbol{X} \in \mathbb{S}^d_{++}$, we define the following mapping*

$$f(\boldsymbol{X}, \boldsymbol{L}_1, \ldots, \boldsymbol{L}_n) = \frac{1}{n}\sum_{i=1}^{n} \lambda_{\max}(\boldsymbol{L}_i) \cdot \lambda_{\max}\left(\boldsymbol{L}_i \boldsymbol{X}^{-1}\right) \cdot \lambda_{\max}\left(\boldsymbol{X}^{-1}\right).$$

Then the above mapping is monotone decreasing in $\boldsymbol{X}$.

Proof. First we notice that from Fact 3 the mapping $\boldsymbol{X} \mapsto \boldsymbol{X}^{-1}$ is monotone decreasing. The latter means that if we have any $\boldsymbol{X}_1, \boldsymbol{X}_2 \in \mathbb{S}^d_{++}$ such that $\boldsymbol{X}_1 \succeq \boldsymbol{X}_2$, we have

$$\boldsymbol{X}_1^{-1} \preceq \boldsymbol{X}_2^{-1}.$$

Then it immediately follows, due to Fact 1, that

$$0 < \lambda_{\max}(\boldsymbol{X}_1^{-1}) \leq \lambda_{\max}(\boldsymbol{X}_2^{-1}).$$

We also notice that the relation $\lambda_{\max}\left(\boldsymbol{L}_i \boldsymbol{X}^{-1}\right) = \lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}} \boldsymbol{X}^{-1} \boldsymbol{L}_i^{\frac{1}{2}}\right) = \lambda_{\max}\left(\boldsymbol{X}^{-1}\boldsymbol{L}_i\right)$, and that the mapping $\boldsymbol{X} \mapsto \boldsymbol{L}_i^{\frac{1}{2}} \boldsymbol{X}^{-1} \boldsymbol{L}_i^{\frac{1}{2}}$ is also monotone decreasing for every $i \in [n]$, so we have

$$0 < \lambda_{\max}\left(\boldsymbol{L}_i \boldsymbol{X}_1^{-1}\right) \leq \lambda_{\max}\left(\boldsymbol{L}_i \boldsymbol{X}_2^{-1}\right).$$

Since we have the coefficient $\lambda_{\max}\left(\boldsymbol{L}_i\right) > 0$, it follows that,

$$f(\boldsymbol{X}_1, \boldsymbol{L}_1, \ldots, \boldsymbol{L}_n) \leq f(\boldsymbol{X}_2, \boldsymbol{L}_1, \ldots, \boldsymbol{L}_n).$$

This means that $f(\boldsymbol{X})$ is monotone decreasing in $\boldsymbol{X}$. □

Fact 5. *For any two matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{S}^d_{++}$, the following relation regarding their largest eigenvalue holds*

$$\lambda_{\max}\left(\boldsymbol{A}\boldsymbol{B}\right) \leq \lambda_{\max}\left(\boldsymbol{A}\right) \cdot \lambda_{\max}\left(\boldsymbol{B}\right). \tag{21}$$

Proof. Using the Courant-Fischer theorem, we can write

$$\begin{aligned}\lambda_{\max}(\boldsymbol{AB}) &= \min_{S:\dim S=d}\max_{x\in S\setminus\{0\}}\frac{x^\top \boldsymbol{AB}x}{x^\top x}\\ &= \max_{x\in\mathbb{R}^d\setminus\{0\}}\frac{x^\top \boldsymbol{AB}x}{x^\top x}\\ &\le \max_{x\in\mathbb{R}^d\setminus\{0\}}\frac{x^\top \boldsymbol{A}x}{x^\top x}\cdot\max_{x\in\mathbb{R}^d\setminus\{0\}}\frac{x^\top \boldsymbol{B}x}{x^\top x}\\ &= \lambda_{\max}(\boldsymbol{A})\cdot\lambda_{\max}(\boldsymbol{B}).\end{aligned}$$

□

Fact 6. *Given matrix $\boldsymbol{Q}\in\mathbb{S}^d_{++}$ and its matrix norm $\|\cdot\|_{\boldsymbol{Q}}$, its associated dual norm is $\|\cdot\|_{\boldsymbol{Q}^{-1}}$.*

Proof. Let $\|\cdot\|_*$ denote the dual norm, then for any vector $z\in\mathbb{R}^d$, we have

$$\|z\|_* = \sup\{z^\top x : \|x\|_{\boldsymbol{Q}}\le 1\}.$$

This is the same as solving $\sup\{z^\top x : \|x\|^2_{\boldsymbol{Q}}=1\}$. The Lagrange function is given as

$$f(x,\lambda) = z^\top x - \lambda\left(\|x\|^2_{\boldsymbol{Q}}-1\right) = z^\top x - \lambda\left(x^\top\boldsymbol{Q}x-1\right).$$

Computing the derivatives we deduce that

$$\frac{\partial f(x,\lambda)}{\partial x} = z - 2\lambda\cdot\boldsymbol{Q}x = 0, \qquad \frac{\partial f(x,\lambda)}{\partial\lambda} = \|x\|^2_{\boldsymbol{Q}}-1=0.$$

This leads to

$$\lambda = \frac{\|z\|_{\boldsymbol{Q}^{-1}}}{2}, \qquad x = \frac{\boldsymbol{Q}^{-1}z}{\|z\|_{\boldsymbol{Q}^{-1}}}.$$

As a result, we have

$$\begin{aligned}\sup\{z^\top x : \|x\|_{\boldsymbol{Q}}\le 1\} &= \sup\{z^\top x : \|x\|^2_{\boldsymbol{Q}}=1\}\\ &= z^\top z = \frac{z^\top\boldsymbol{Q}^{-1}z}{\|z\|_{\boldsymbol{Q}^{-1}}}\\ &= \|z\|_{\boldsymbol{Q}^{-1}}.\end{aligned}$$

This completes the proof. □

B Background and Motivation

Ideally, for any given $\varepsilon>0$, the goal is to find a point $x_\varepsilon\in\mathbb{R}^d$ satisfying: $f(x)-f^\star\ge\varepsilon$. However, it's important to note that in the general case, finding such an approximately global optimum is known to be NP-hard, as discussed in [JK+17, DDG+22]. On the contrary, convex optimization problems are well-studied, and there exists an extensive body of literature on various methods. In the context of our work, methods based on gradient descent are of particular interest. When these methods are applied to non-convex objectives, they treat the function f as locally convex and aim to converge to a local minimum. Despite this simplification, such methods have gained popularity in practice due to their superior performance compared to other approaches for non-convex optimization, such as convex relaxation-based methods [Tib96, CCS10].

Stochastic Gradient Descent. Arguably, one of the most prominent meta-methods for tackling non-convex optimization problems is stochastic gradient descent (SGD). The formulation of SGD is presented as the following iterative algorithm:

$$x^{k+1} = x^k - \gamma g^k. \tag{SGD}$$

Here, $g^k \in \mathbb{R}^d$ serves as a stochastic estimator of the gradient $\nabla f(x^k)$. SGD essentially mimics the classical gradient descent algorithm, and recovers it when $g^k = \nabla f(x^k)$. In this scenario, the method approximates the objective function f using a linear function and takes a step of size γ in the direction that maximally reduces this approximation. When the stepsize is sufficiently small, and the function f is suitably smooth, it can be demonstrated that the function value decreases, as discussed in [B$^+$15, GLQ$^+$19].

However, computing the full gradient can often be computationally expensive. In such cases, stochastic approximations of the gradient come into play. Stochastic estimators of the gradient can be employed for various purposes, leading to the development of different methods. These include stochastic batch gradient descent [NJLS09, JZ13, DBLJ14], randomized coordinate descent [Nes12, Wri15], and compressed gradient descent [AGL$^+$17, KFJ18, MGTR19]. The latter, compressed gradient descent, holds particular relevance to this paper, and we will delve into a more detailed discussion of it in subsequent sections.

Second order methods. The stochastic gradient descent is considered as a first-order method as it uses only the first order derivative information. Although being immensely popular, the first order methods are not always the most optimal. Not surprisingly, using higher order derivatives in deciding update direction can yield to faster algorithms. A simple instance of such algorithms is the Newton Star algorithm [IQR21]:

$$x^{k+1} = x^k - \left(\nabla^2 f(x^\star)\right)^{-1} \nabla f(x^k), \tag{NS}$$

where $x^\star$ is the minimum point of the objective function. The authors establish that under specific conditions, the algorithm's convergence to the unique solution $x^\star$ in the convex scenario occurs at a local quadratic rate. Nonetheless, its practicality is limited since we do not have prior knowledge of the Hessian matrix at the optimal point. Despite being proposed recently, the Newton-Star algorithm gives a deeper insight on the generic Newton method [GT74, Mie80, Yam87]:

$$x^{k+1} = x^k - \gamma \left(\nabla^2 f(x^k)\right)^{-1} \nabla f(x^k). \tag{NM}$$

Here, the unknown Hessian of the Newton-Star algorithm, is estimated progressively along the iterations. The latter causes elevated computational costs, as the inverting a large square matrix is expensive. As an alternative, quasi-Newton methods replace the inverse of the Hessian at the iterate with a computationally cheaper estimate [Bro65, DM77, ABK07, ABSM14].

Fixed matrix stepsizes. The `det-CGD` algorithm falls into this framework of the second order methods as well. Proposed by [LKR23][2], the algorithm suggests using a uniform "upper bound" on the inverse Hessian matrix. Assuming matrix smoothness of the objective [SHR21], they replace the scalar stepsize with a positive definite matrix $\boldsymbol{D}$. The algorithm is given as follows:

$$x^{k+1} = x^k - \boldsymbol{D}\boldsymbol{S}^k \nabla f(x^k). \tag{det-CGD}$$

- Here, $\boldsymbol{D}$ plays the role of the stepsize. It essentially uniformly upper bounds the inverse Hessian. The standard SGD is a particular case of this method, as the scalar stepsize γ can be seen as a matrix $\gamma \boldsymbol{I}_d$, where $\boldsymbol{I}_d$ is the d-dimensional identity matrix. An advantage of using a matrix stepsize is more evident if we take the perspective of the second order methods. Indeed, the scalar stepsize $\gamma \boldsymbol{I}_d$ uniformly estimates the largest eigenvalue of the Hessian matrix, while $\boldsymbol{D}$ can capture the Hessian more accurately. The authors show both theoretical and empirical improvement that comes with matrix stepsizes.
- $\boldsymbol{S}^k$ is a positive semi-definite stochastic sketch matrix, that is unbiased: $\mathbb{E}[\boldsymbol{S}^k] = \boldsymbol{I}_d$. We notice that `det-CGD` can be seen as a matrix stepsize instance of `SGD`, with $g^k = \boldsymbol{S}^k \nabla f(x^k)$. The sketch matrix can be seen as a linear compressing operator, hence the name of the algorithm: CGD (Compressed Gradient Descent) (see [AGL$^+$17, KFJ18]). A commonly used example of such a compressor is the Rand-k compressor. This compressor randomly selects m entries from its input and scales them using a scalar multiplier to ensure an

[2] In the original paper, the algorithm is referred to as `det-CGD`, as there is a variant of the same algorithm named `det-CGD2`. Since we are going to use only the first one and our framework is applicable to both, we will remove the number in the end for the sake of brevity.

unbiased estimation. By adopting this approach, instead of using all d coordinates of the gradient, only a subset of size m is communicated. Formally, rand-τ is defined as follows:

$$\boldsymbol{S} = \frac{d}{\tau} \sum_{j=1}^{\tau} e_{i_j} e_{i_j}^{\top}. \tag{22}$$

Here, e_{i_j} denotes the i_j-th standard basis vector in $\mathbb{R}^d$. For a more comprehensive understanding of compression techniques, we refer to the paper by [SSR22].

The neighborhood of the distributed det-CGD1. The distributed version of `det-CGD` follows the standard federated learning paradigm [MMR+17]. The pseudocode of the method, as well as the convergence result of [LKR23], can be found in the appendix. Informally, their convergence result can be written as

$$\min_{k=1,\ldots,K} \mathbb{E}[\left\|\nabla f(x^k)\right\|_{\boldsymbol{D}}^2] \leq \mathcal{O}\left(\frac{(1+\alpha)^K}{K}\right) + \mathcal{O}(\alpha), \tag{23}$$

where $\alpha > 0$ is a constant that can be controlled. The crucial insight from this result is that the error bound doesn't diminish as the number of iterations increases. In fact, by controlling α and considering a large K, it's impossible to make the second term smaller than ε. This implies that the algorithm converges to a certain neighborhood surrounding the (local) optimum. This phenomenon is a common occurrence in `SGD` and is primarily attributable to the variance associated with the stochastic gradient estimator. In the case of `det-CGD` the stochasticity comes from the sketch $\boldsymbol{S}^k$.

Variance reduction. To eliminate this neighborhood, various techniques for reducing variance are employed. One of the simplest techniques applicable to `CGD` is gradient shifting. By replacing $\boldsymbol{S}^k \nabla f(x^k)$ with $\boldsymbol{S}^k(\nabla f(x^k) - \nabla f(x^\star)) + \nabla f(x^\star)$, the neighborhood effect is removed from the general `CGD`. This algorithm is an instance of a more commonly known method called $\texttt{SGD}_\star$ [GSBR20]. However, since the exact optimum $x^\star$ is typically unknown, this technique encounters similar challenges as the Newton-Star algorithm mentioned earlier. Fortunately, akin to quasi-Newton methods, one can employ methods that iteratively learn the optimal shift [SR22].

A line of research focuses on variance reduction for `CGD` based on this insight. To mitigate the neighborhood effect in the distributed version of `CGD`, denoted as `det-CGD1`, we apply a technique called `MARINA` [GBLR21]. `MARINA` cleverly combines the general shifting technique with loopless variance reduction techniques [QQR21]. This approach introduces an alternative gradient estimator specifically designed for the federated learning setting. Thanks to its structure, it allows to establish an upper bound on the stationarity error that diminishes significantly with a large number of iterations. In this paper, we construct the analog of the this algorithm called `det-MARINA`, using matrix stepsizes and sketch gradient compressors. For this new method, we prove a convergence guarantee similar to (23) that without a neighborhood term.

C Prior Work

Numerous effective convex optimization techniques have been adapted for application in non-convex scenarios. Here's a selection of these techniques, although it's not an exhaustive list: adaptivity [DOG+19, ZKV+20], variance reduction [JRSPS16, LBZR21], and acceleration [GNDG19]. Of particular relevance to our work is the paper by [KR20], which introduces a unified approach for analyzing stochastic gradient descent for non-convex objectives. A comprehensive overview of non-convex optimization can be found in [JK+17, DDG+22].

An illustrative example of a matrix stepsized method is Newton's method, which has been a long-standing favorite in the optimization community [GT74, Mie80, Yam87]. However, the computational complexity involved in computing the stepsize as the inverse of the Hessian of the current iteration is substantial. Instead, quasi-Newton methods employ a readily computable estimator to replace the inverse Hessian [Bro65, DM77, ABK07, ABSM14]. An important direction of research that is relevant to our work, studies distributed second order methods. Here is a non-exhaustive list of papers in this area: [WRXM18, CR19, ZYB20, IQR21, ADA21, SIQR22].

The Distributed Compressed Gradient Descent (`DCGD`) algorithm, initially proposed by [KFJ18], has seen improvements in various aspects, as documented in works such as [LKQR20, HHH+22]. Its

variance reduced version with gradients shifts was studied by [SR22] in the (strongly) convex setting. Additionally, there exists a substantial body of literature on other federated learning algorithms employing unbiased compressors [AGL$^+$17, MGTR19, GBLR21, MMSR22, MSR22, HKM$^+$23].

Variance reduction techniques have gained significant attention in the context of stochastic batch gradient descent that is prevalent in machine learning. Numerous algorithms have been developed in this regard, including well-known ones like `SVRG` [JZ13], `SAG` [SLRB17], `SDCA` [RT14], `SAGA` [DBLJ14], `MISO` [Mai15], and `Katyusha` [AZ17]. An overview of more advanced methods can be found in [GSBR20]. Notably, `SVRG` and `Katyusha` have been extended with loopless variants, namely `L-SVRG` and `L-Katyusha` [KHR20, QQR21]. These loopless versions streamline the algorithms by eliminating the outer loop and introducing a biased coin-flip mechanism at each step. This simplification eases both the algorithms' structure and their analyses, while preserving their worst-case complexity bounds. `L-SVRG`, in particular, offers the advantage of setting the exit probability from the outer loop independently of the condition number, thus, enhancing both robustness and practical efficiency.

This technique of coin flipping allows to obtain variance reduction for the `CGD` algorithm. A relevant example is the `DIANA` algorithm proposed by [MGTR19]. Its convergence was proved both in the convex and non-convex cases. Later, `MARINA` [GBLR21] obtained the optimal convergence rates, improving in communication complexity compared to all previous first order methods. Finally, there is a line of work developing variance reduction in the federated setting using other methods and techniques [CKK$^+$19, HR20, DTN$^+$20, PWLC22, TR22].

D Properties of Matrix Smoothness

D.1 The Matrix Lipschitz-Continuous Gradient

In this section we describe some properties of matrix smoothness, matrix gradient Lipschitzness and their relations.

D.1.1 Quadratics

Given a matrix $\boldsymbol{A} \in \mathbb{S}^d_{++}$ and a vector $b \in \mathbb{R}^d$, consider the function $f(x) = \frac{1}{2}x^\top \boldsymbol{A}x + b^\top x + c$. Then its gradient is computed as $\nabla f(x) = \boldsymbol{A}x + b$ and $\nabla^2 f(x) = \boldsymbol{A}$. Inserting gradients formula into (4) we deduce

$$\sqrt{(x-y)^\top \boldsymbol{A}\boldsymbol{L}^{-1}\boldsymbol{A}(x-y)} \leq \sqrt{(x-y)^\top \boldsymbol{L}(x-y)},$$

for any $x, y \in \mathbb{R}^d$. This reduces to

$$\boldsymbol{A}\boldsymbol{L}^{-1}\boldsymbol{A} \preceq \boldsymbol{L}. \tag{24}$$

Since $\boldsymbol{A} \in \mathbb{S}^d_{++}$, we can also rewrite (24) as

$$\boldsymbol{A}^{\frac{1}{2}}\boldsymbol{L}^{-1}\boldsymbol{A}^{\frac{1}{2}} \preceq \boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{L}\boldsymbol{A}^{-\frac{1}{2}},$$

which is equivalent to

$$\boldsymbol{A} \preceq \boldsymbol{L}. \tag{25}$$

Therefore, the "best" $\boldsymbol{L} \in \mathbb{S}^d_{++}$ that satisfies (4) is $\boldsymbol{L} = \boldsymbol{A} = \nabla^2 f(x)$, for every $x \in \mathbb{R}^d$. Now, let us look at a more general setting. Consider f given as follows,

$$f(x) = \sum_{i=1}^{s} \phi_i(\boldsymbol{M}_i x),$$

where $\boldsymbol{M}_i \in \mathbb{R}^{q_i \times d}$. Here $f : \mathbb{R}^d \mapsto \mathbb{R}$ is the sum of functions $\phi_i : \mathbb{R}^{q_i} \mapsto \mathbb{R}$. We assume that each function ϕ_i has matrix $\boldsymbol{L}_i$ Lipschitz gradient. We have the following lemma regarding the matrix gradient Lipschitzness of f.

Proposition 3. *Assume that functions f and $\{\phi_i\}_{i=1}^s$ are described above. Then function f has $\boldsymbol{L}$-Lipschitz gradient, if the following condition is satisfied:*

$$\sum_{i=1}^{s} \lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}}\boldsymbol{M}_i\boldsymbol{L}^{-1}\boldsymbol{M}_i^\top\boldsymbol{L}_i^{\frac{1}{2}}\right) = 1. \tag{26}$$

Note that Proposition 3 is a generalization of the previous case of quadratics, if we pick $s = 1$, $\boldsymbol{M}_i = \boldsymbol{A}^{\frac{1}{2}}$ and $\phi_1(x) = x^\top \boldsymbol{I}_d x$, the condition becomes $\boldsymbol{L} = \boldsymbol{A}$, which is exactly the solution given by (25). Thus we recover the result for quadratics. The linear term $bx + c$ is ignored in this case. In Proposition 3, we only intend to give a way of finding a matrix $\boldsymbol{L} \in \mathbb{S}^d_{++}$, so that f has $\boldsymbol{L}$-Lipschitz gradient. This does not mean, however, the $\boldsymbol{L}$ here is optimal. The proof is deferred to Appendix D.3.

D.2 Comparison of the Different Smoothness Conditions

We provide a proposition here which describes an equivalent form of stating $\boldsymbol{L}$-matrix smoothness of a function f. This proposition is used to illustrate the relation between matrix smoothness and matrix Lipschitz gradient.

Proposition 4. *Let function $f : \mathbb{R}^d \to \mathbb{R}$ be continuously differentiable. Then the following statements are equivalent.*

(i) f is $\boldsymbol{L}$-matrix smooth.

(ii) $\langle \nabla f(x) - \nabla f(y), x - y \rangle \leq \|x - y\|^2_{\boldsymbol{L}}$ for all $x, y \in \mathbb{R}^d$.

The two propositions, Proposition 5 and Proposition 6, formulated below illustrate the relation between matrix smoothness of f and matrix gradient Lipschitzness of f.

Proposition 5. *Assume $f : \mathbb{R}^d \mapsto \mathbb{R}$ is a continuously differentiable function, and its gradient is $\boldsymbol{L}$-Lipschitz continuous with $\boldsymbol{L} \in \mathbb{S}^d_{++}$. Then function f is $\boldsymbol{L}$-matrix smooth.*

Proposition 6. *Assume $f : \mathbb{R}^d \to \mathbb{R}$ is a continuously differentiable function. Assume also that f is convex and $\boldsymbol{L}$-matrix smooth. Then ∇f is $\boldsymbol{L}$-Lipschitz continuous.*

The next proposition shows that standard Lipschitzness of the gradient of a function is an immediate consequence of matrix Lipschitzness.

Proposition 7. *Assume that the gradient of f is $\boldsymbol{L}$-Lipschitz continuous. Then ∇f is also L-Lipschitz with $L = \lambda_{\max}(\boldsymbol{L})$.*

D.3 Proofs of the Propositions Regarding Smoothness

D.3.1 Proof of Proposition 1

We start with picking any two vector $x, y \in \mathbb{R}^d$. We have

$$\begin{aligned}
&\|\nabla f(x) - \nabla f(y)\|^2_{\boldsymbol{L}^{-1}} \\
&= \left\| \int_0^1 \nabla^2 f(\theta x + (1-\theta)y)(x - y)\, \mathrm{d}\theta \right\|^2_{\boldsymbol{L}^{-1}} \\
&= (x-y)^\top \left(\int_0^1 \nabla^2 f(\theta x + (1-\theta)y)\, \mathrm{d}\theta \right)^\top \boldsymbol{L}^{-1} \left(\int_0^1 \nabla^2 f(\theta x + (1-\theta)y)\, \mathrm{d}\theta \right)(x - y).
\end{aligned}$$

Denote $\boldsymbol{F} := \int_0^1 \nabla^2 f(\theta x + (1-\theta)y)\, \mathrm{d}\theta$, notice that $\boldsymbol{F}$ is a symmetric matrix. Then, the previous identity becomes

$$\|\nabla f(x) - \nabla f(y)\|^2_{\boldsymbol{L}^{-1}} = (x-y)^\top \boldsymbol{F}^\top \boldsymbol{L}^{-1} \boldsymbol{F}(x - y).$$

From the definition of $\boldsymbol{F}$ and the bounded Hessian assumption, we have $\boldsymbol{F} \preceq \boldsymbol{L}$. Let us prove that $\boldsymbol{F}\boldsymbol{L}^{-1}\boldsymbol{F} \preceq \boldsymbol{L}$:

$$\begin{aligned}
\boldsymbol{F}\boldsymbol{L}^{-1}\boldsymbol{F} \preceq \boldsymbol{L} \quad &\iff \quad \boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{F}\boldsymbol{L}\boldsymbol{F}\boldsymbol{L}^{-\frac{1}{2}} \preceq \boldsymbol{I}_d \\
&\iff \quad \boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{F}\boldsymbol{L}^{-\frac{1}{2}} \cdot \boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{F}\boldsymbol{L}^{-\frac{1}{2}} \preceq \boldsymbol{I}_d \\
&\iff \quad \boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{F}\boldsymbol{L}^{-\frac{1}{2}} \preceq \boldsymbol{I}_d \\
&\iff \quad \boldsymbol{F} \preceq \boldsymbol{L}.
\end{aligned}$$

This means that

$$\begin{aligned}
\|\nabla f(x) - \nabla f(y)\|^2_{\boldsymbol{L}^{-1}} &\leq (x-y)^\top \boldsymbol{L}(x - y) \\
&= \|x - y\|^2_{\boldsymbol{L}},
\end{aligned}$$

which completes the proof.

D.3.2 Proof of Proposition 2

We start with the definition of $\boldsymbol{L}$-Lipschitz gradient of function f, and pick two arbitrary points $x, y \in \mathbb{R}^d$,

$$\|\nabla f(x) - \nabla f(y)\|_{\boldsymbol{L}^{-1}}^2 \quad = \quad \left\| \frac{1}{n} \sum_{i=1}^n (\nabla f_i(x) - \nabla f_i(y)) \right\|_{\boldsymbol{L}^{-1}}^2 .$$

Applying the convexity of $\|\cdot\|_{\boldsymbol{L}^{-1}}^2$, we have

$$\|\nabla f(x) - \nabla f(y)\|_{\boldsymbol{L}^{-1}}^2 \quad \leq \quad \frac{1}{n} \sum_{i=1}^n \|\nabla f_i(x) - \nabla f_i(y)\|_{\boldsymbol{L}^{-1}}^2 .$$

For each term within the summation, we use the definition of matrix norms and replace the matrix $\boldsymbol{L}^{-1}$ with $\boldsymbol{L}_i^{-1/2} \boldsymbol{L}_i^{1/2} \boldsymbol{L}^{-1} \boldsymbol{L}_i^{1/2} \boldsymbol{L}_i^{-1/2}$, for every $i = 1, \ldots, n$:

$$\begin{aligned}
\|\nabla f(x) - \nabla f(y)\|_{\boldsymbol{L}^{-1}}^2 &= \frac{1}{n} \sum_{i=1}^n \left(\boldsymbol{L}_i^{-\frac{1}{2}} (\nabla f_i(x) - \nabla f_i(y)) \right)^\top \boldsymbol{L}_i^{\frac{1}{2}} \boldsymbol{L}^{-1} \boldsymbol{L}_i^{\frac{1}{2}} \left(\boldsymbol{L}_i^{-\frac{1}{2}} (\nabla f_i(x) - \nabla f_i(y)) \right) \\
&\leq \frac{1}{n} \sum_{i=1}^n \lambda_{\max} \left(\boldsymbol{L}_i^{\frac{1}{2}} \boldsymbol{L}^{-1} \boldsymbol{L}_i^{\frac{1}{2}} \right) \left\| \boldsymbol{L}_i^{-\frac{1}{2}} (\nabla f_i(x) - \nabla f_i(y)) \right\|^2 \\
&= \frac{1}{n} \sum_{i=1}^n \lambda_{\max} \left(\boldsymbol{L}_i^{\frac{1}{2}} \boldsymbol{L}^{-1} \boldsymbol{L}_i^{\frac{1}{2}} \right) \|\nabla f_i(x) - \nabla f_i(y)\|_{\boldsymbol{L}_i^{-1}}^2 .
\end{aligned}$$

Using the assumption that the gradient of each function f_i is $\boldsymbol{L}_i$-Lipschitz, we obtain,

$$\|\nabla f(x) - \nabla f(y)\|_{\boldsymbol{L}^{-1}}^2 \quad \leq \quad \frac{1}{n} \sum_{i=1}^n \lambda_{\max} \left(\boldsymbol{L}_i^{\frac{1}{2}} \boldsymbol{L}^{-1} \boldsymbol{L}_i^{\frac{1}{2}} \right) \|x - y\|_{\boldsymbol{L}_i}^2 .$$

We apply the same replacement trick again, this time replacing $\boldsymbol{L}_i^{-1}$ with $\boldsymbol{L}^{-1/2} \boldsymbol{L}^{1/2} \boldsymbol{L}_i^{-1} \boldsymbol{L}^{1/2} \boldsymbol{L}^{-1/2}$

$$\begin{aligned}
\|\nabla f(x) - \nabla f(y)\|_{\boldsymbol{L}^{-1}}^2 &= \frac{1}{n} \sum_{i=1}^n \lambda_{\max} \left(\boldsymbol{L}_i^{\frac{1}{2}} \boldsymbol{L}^{-1} \boldsymbol{L}_i^{\frac{1}{2}} \right) \cdot \left[(\boldsymbol{L}^{\frac{1}{2}} (x - y))^\top \boldsymbol{L}^{-\frac{1}{2}} \boldsymbol{L}_i \boldsymbol{L}^{-\frac{1}{2}} (\boldsymbol{L}^{\frac{1}{2}} (x - y)) \right] \\
&\leq \frac{1}{n} \sum_{i=1}^n \lambda_{\max} \left(\boldsymbol{L}_i^{\frac{1}{2}} \boldsymbol{L}^{-1} \boldsymbol{L}_i^{\frac{1}{2}} \right) \cdot \lambda_{\max} \left(\boldsymbol{L}^{-\frac{1}{2}} \boldsymbol{L}_i \boldsymbol{L}^{-\frac{1}{2}} \right) \left\| \boldsymbol{L}^{\frac{1}{2}} (x - y) \right\|^2 .
\end{aligned}$$

Using Fact 5, we are deduce the following bound,

$$\begin{aligned}
\|\nabla f(x) - \nabla f(y)\|_{\boldsymbol{L}^{-1}}^2 &\leq \left(\frac{1}{n} \sum_{i=1}^n \lambda_{\max} \left(\boldsymbol{L}^{-1} \right) \cdot \lambda_{\max} \left(\boldsymbol{L}_i \right) \cdot \lambda_{\max} \left(\boldsymbol{L}_i \boldsymbol{L}^{-1} \right) \right) \cdot \|x - y\|_{\boldsymbol{L}}^2 \\
&= \|x - y\|_{\boldsymbol{L}}^2 .
\end{aligned}$$

This completes the proof.

D.3.3 Proof of Proposition 3

For any x and y from $\mathbb{R}^d$, we have

$$\begin{aligned}
&\|\nabla f(x) - \nabla f(y)\|_{\boldsymbol{L}^{-1}} \\
&= \left\| \sum_{i=1}^s \boldsymbol{M}_i^\top \nabla \phi_i(\boldsymbol{M}_i x) - \sum_{i=1}^s \boldsymbol{M}_i^\top \nabla \phi_i(\boldsymbol{M}_i y) \right\|_{\boldsymbol{L}^{-1}} \\
&= s \cdot \left\| \frac{1}{s} \sum_{i=1}^s \boldsymbol{M}_i^\top \left(\nabla \phi_i(\boldsymbol{M}_i x) - \nabla \phi_i \left(\boldsymbol{M}_i y \right) \right) \right\|_{\boldsymbol{L}^{-1}} .
\end{aligned}$$

Applying the convexity of the norm $\|\cdot\|_{\boldsymbol{L}^{-1}}$,

$$\|\nabla f(x) - \nabla f(y)\|_{\boldsymbol{L}^{-1}} \quad \leq \quad s \cdot \frac{1}{s} \sum_{i=1}^{s} \left\| \boldsymbol{M}_i^\top \left(\nabla \phi_i(\boldsymbol{M}_i x) - \nabla \phi_i(\boldsymbol{M}_i y) \right) \right\|_{\boldsymbol{L}^{-1}}.$$

Expanding the norm and applying the replacement trick for above $\boldsymbol{L}$ and $\boldsymbol{M}_i$, we obtain

$$\begin{aligned}
&\|\nabla f(x) - \nabla f(y)\|_{\boldsymbol{L}^{-1}} \\
&= \sum_{i=1}^{s} \sqrt{\left(\nabla \phi_i(\boldsymbol{M}_i x) - \nabla \phi_i(\boldsymbol{M}_i y) \right)^\top \boldsymbol{M}_i \boldsymbol{L}^{-1} \boldsymbol{M}_i^\top \left(\nabla \phi_i(\boldsymbol{M}_i x) - \nabla \phi_i(\boldsymbol{M}_i y) \right)} \\
&= \sum_{i=1}^{s} \sqrt{\left[\boldsymbol{L}_i^{-\frac{1}{2}} \left(\nabla \phi_i(\boldsymbol{M}_i x) - \nabla \phi_i(\boldsymbol{M}_i y) \right) \right]^\top \boldsymbol{L}_i^{\frac{1}{2}} \boldsymbol{M}_i \boldsymbol{L}^{-1} \boldsymbol{M}_i^\top \boldsymbol{L}_i^{\frac{1}{2}} \left[\boldsymbol{L}_i^{-\frac{1}{2}} \left(\nabla \phi_i(\boldsymbol{M}_i x) - \nabla \phi_i(\boldsymbol{M}_i y) \right) \right]} \\
&\leq \sum_{i=1}^{s} \sqrt{\lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}} \boldsymbol{M}_i \boldsymbol{L}^{-1} \boldsymbol{M}_i^\top \boldsymbol{L}_i^{\frac{1}{2}} \right)} \cdot \|\nabla \phi_i(\boldsymbol{M}_i x) - \nabla \phi_i(\boldsymbol{M}_i y)\|_{\boldsymbol{L}_i^{-1}}.
\end{aligned}$$

Due to the assumption that the gradient of ϕ_i is $\boldsymbol{L}_i$-Lipschitz, we have

$$\begin{aligned}
&\|\nabla f(x) - \nabla f(y)\|_{\boldsymbol{L}^{-1}} \\
&\leq \sum_{i=1}^{s} \sqrt{\lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}} \boldsymbol{M}_i \boldsymbol{L}^{-1} \boldsymbol{M}_i^\top \boldsymbol{L}_i^{\frac{1}{2}} \right)} \cdot \|\boldsymbol{M}_i(x - y)\|_{\boldsymbol{L}_i} \\
&= \sum_{i=1}^{s} \sqrt{\lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}} \boldsymbol{M}_i \boldsymbol{L}^{-1} \boldsymbol{M}_i^\top \boldsymbol{L}_i^{\frac{1}{2}} \right)} \cdot \sqrt{\left[\boldsymbol{L}^{\frac{1}{2}} (x - y) \right]^\top \boldsymbol{L}^{-\frac{1}{2}} \boldsymbol{M}_i^\top \boldsymbol{L}_i \boldsymbol{M}_i \boldsymbol{L}^{-\frac{1}{2}} \left[\boldsymbol{L}^{\frac{1}{2}} (x - y) \right]} \\
&\leq \sum_{i=1}^{s} \sqrt{\lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}} \boldsymbol{M}_i \boldsymbol{L}^{-1} \boldsymbol{M}_i^\top \boldsymbol{L}_i^{\frac{1}{2}} \right) \cdot \lambda_{\max}\left(\boldsymbol{L}^{-\frac{1}{2}} \boldsymbol{M}_i^\top \boldsymbol{L}_i \boldsymbol{M}_i \boldsymbol{L}^{-\frac{1}{2}} \right)} \cdot \|x - y\|_{\boldsymbol{L}} \\
&\leq \sum_{i=1}^{s} \lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}} \boldsymbol{M}_i \boldsymbol{L}^{-1} \boldsymbol{M}_i^\top \boldsymbol{L}_i^{\frac{1}{2}} \right) \cdot \|x - y\|_{\boldsymbol{L}},
\end{aligned}$$

where the last inequality is due to the fact that,

$$\lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}} \boldsymbol{M}_i \boldsymbol{L}^{-1} \boldsymbol{M}_i^\top \boldsymbol{L}_i^{\frac{1}{2}} \right) = \lambda_{\max}\left(\boldsymbol{L}^{-\frac{1}{2}} \boldsymbol{M}_i^\top \boldsymbol{L}_i \boldsymbol{M}_i \boldsymbol{L}^{-\frac{1}{2}} \right).$$

Recalling the condition of the proposition:

$$\sum_{i=1}^{s} \lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}} \boldsymbol{M}_i \boldsymbol{L}^{-1} \boldsymbol{M}_i^\top \boldsymbol{L}_i^{\frac{1}{2}} \right) = 1,$$

we deduce

$$\|\nabla f(x) - \nabla f(y)\|_{\boldsymbol{L}^{-1}} \leq \|x - y\|_{\boldsymbol{L}}.$$

This completes the proof.

D.3.4 Proof of Proposition 4

(i) $\rightarrow$ (ii). If f is $\boldsymbol{L}$-matrix smooth, then for all $x, y \in \mathbb{R}^d$, we have

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{1}{2} \|x - y\|_{\boldsymbol{L}}^2,$$

and

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2} \|x - y\|_{\boldsymbol{L}}^2.$$

Summing up these two inequalities we get

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \leq \|x - y\|_{\boldsymbol{L}}^2.$$

(ii) $\to$ (i). Choose any $x, y \in \mathbb{R}^d$, and define $z = x + t(y - x)$, then we have,

$$\begin{aligned} f(y) &= f(x) + \int_0^1 \langle \nabla f(x + t(y - x)), y - x \rangle \, \mathrm{d}t \\ &= f(x) + \int_0^1 \langle \nabla f(z), y - x \rangle \, \mathrm{d}t \\ &= f(x) + \langle \nabla f(x), y - x \rangle + \int_0^1 \langle \nabla f(z) - \nabla f(x), y - x \rangle \, \mathrm{d}t \\ &= f(x) + \langle \nabla f(x), y - x \rangle + \int_0^1 \langle \nabla f(z) - \nabla f(x), z - x \rangle \cdot \frac{1}{t} \, \mathrm{d}t. \end{aligned}$$

Using the assumption that for any $x, z \in \mathbb{R}^d$, we have

$$\langle \nabla f(z) - \nabla f(x), z - x \rangle \leq \|z - x\|_{\boldsymbol{L}}^2 .$$

Plug this back into the previous identity, we obtain

$$\begin{aligned} f(y) &\leq f(x) + \langle \nabla f(x), y - x \rangle + \int_0^1 \|z - x\|_{\boldsymbol{L}}^2 \cdot \frac{1}{t} \, \mathrm{d}t \\ &= f(x) + \langle \nabla f(x), y - x \rangle + \int_0^1 \|y - x\|_{\boldsymbol{L}}^2 \cdot t \, \mathrm{d}t \\ &= f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2} \|y - x\|_{\boldsymbol{L}}^2 . \end{aligned}$$

We conclude the proof.

D.3.5 Proof of Proposition 5

We start with picking any two points $x, y \in \mathbb{R}^d$, using the generalized Cauchy-Schwarz inequality for dual norm, we have

$$\begin{aligned} \langle \nabla f(x) - \nabla f(y), x - y \rangle &\leq \|\nabla f(x) - \nabla f(y)\|_{\boldsymbol{L}^{-1}} \cdot \|x - y\|_{\boldsymbol{L}} \\ &\overset{(4)}{\leq} \|x - y\|_{\boldsymbol{L}} \cdot \|x - y\|_{\boldsymbol{L}} \\ &= \|x - y\|_{\boldsymbol{L}}^2 \end{aligned}$$

From Proposition 4, this indicates that function f is $\boldsymbol{L}$-matrix smooth. This completes the proof.

D.3.6 Proof of Proposition 6

Using Proposition 4, we know that for any $x, y \in \mathbb{R}^d$, we have

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \leq \|x - y\|_{\boldsymbol{L}}^2 . \tag{27}$$

Now we pick any three points $x, y, z \in \mathbb{R}^d$. Using the $\boldsymbol{L}$-smoothness of f, we have

$$f(x + z) \geq f(x) + \langle \nabla f(x), z \rangle + \frac{1}{2} \|z\|_{\boldsymbol{L}}^2 . \tag{28}$$

Using the convexity of f we have

$$\langle \nabla f(y), x + z - y \rangle \leq f(x + z) - f(y). \tag{29}$$

Combining (28) and (29), we obtain

$$\langle \nabla f(y), x + z - y \rangle \leq f(x) - f(y) + \langle \nabla f(x), z \rangle + \frac{1}{2} \|z\|_{\boldsymbol{L}}^2 .$$

Rearranging terms we get

$$\langle \nabla f(y) - \nabla f(x), z \rangle - \frac{1}{2} \|z\|_{\boldsymbol{L}}^2 \leq f(x) - f(y) - \langle \nabla f(y), x - y \rangle .$$

The inequality holds for any z for fixed x and y, and the left hand side is maximized (w.r.t. z) when $z = \boldsymbol{L}^{-1}(\nabla f(y) - \nabla f(x))$. Plugging it in, we get

$$\frac{1}{2}\|\nabla f(x) - \nabla f(y)\|^2_{\boldsymbol{L}^{-1}} \leq f(x) - f(y) - \langle \nabla f(y), x - y\rangle . \tag{30}$$

By symmetry we can also obtain

$$\frac{1}{2}\|\nabla f(y) - \nabla f(x)\|^2_{\boldsymbol{L}^{-1}} \leq f(y) - f(x) - \langle \nabla f(x), y - x\rangle .$$

Adding (30) and its counterpart together, we get

$$\|\nabla f(x) - \nabla f(y)\|^2_{\boldsymbol{L}^{-1}} \leq \langle \nabla f(x) - \nabla f(y), x - y\rangle . \tag{31}$$

Combing (31) and (27), it follows

$$\|\nabla f(x) - \nabla f(y)\|^2_{\boldsymbol{L}^{-1}} \leq \|x - y\|^2_{\boldsymbol{L}} .$$

Note that $\boldsymbol{L}$ and $\boldsymbol{L}^{-1}$ are both positive definite matrices, so it is equivalent to

$$\|\nabla f(x) - \nabla f(y)\|_{\boldsymbol{L}^{-1}} \leq \|x - y\|_{\boldsymbol{L}} .$$

This completes the proof.

D.3.7 Proof of Proposition 7

Let us start with picking any two points $x, y \in \mathbb{R}^d$. With the matrix $\boldsymbol{L}$-Lipschitzness of the gradient of function f, we have

$$\|\nabla f(x) - \nabla f(y)\|^2_{\boldsymbol{L}^{-1}} \leq \|x - y\|^2_{\boldsymbol{L}} .$$

This implies

$$(x - y)^\top \boldsymbol{L}(x - y) - (\nabla f(x) - \nabla f(y))^\top \boldsymbol{L}^{-1}(\nabla f(x) - \nabla f(y)) \geq 0.$$

Define function $f(\boldsymbol{X}) := a^\top \boldsymbol{X} a - b^\top \boldsymbol{X}^{-1} b$ for $\boldsymbol{X} \in \mathbb{S}^d_{++}$, where $a, b \in \mathbb{R}^d$ are fixed vectors. Then f is monotone increasing in $\boldsymbol{X}$. This can be shown in the following way, picking two matrices $\boldsymbol{X}_1, \boldsymbol{X}_2 \in \mathbb{S}^d_{++}$, where $\boldsymbol{X}_1 \succeq \boldsymbol{X}_2$. It is easy to see that $-\boldsymbol{X}_1^{-1} \succeq -\boldsymbol{X}_2^{-1}$, since from Fact 3 the map $\boldsymbol{X} \mapsto -\boldsymbol{X}^{-1}$ is monotone increasing for $\boldsymbol{X} \in \mathbb{S}^d_{++}$. Thus,

$$\begin{aligned} f(\boldsymbol{X}_1) - f(\boldsymbol{X}_2) &= (x - y)^\top (\boldsymbol{X}_1 - \boldsymbol{X}_2)(x - y) \\ &\quad + (\nabla f(x) - \nabla f(y))^\top \left(-\boldsymbol{X}_1^{-1} - (-\boldsymbol{X}_2^{-1})\right)(\nabla f(x) - \nabla f(y)) \\ &\geq 0. \end{aligned}$$

As a result, $f(\lambda_{\max}(\boldsymbol{L}) \cdot \boldsymbol{I}_d) \geq f(\boldsymbol{L}) \geq 0$, due to the fact that $\lambda_{\max}(\boldsymbol{L}) \cdot \boldsymbol{I}_d \succeq \boldsymbol{L}$. It remains to notice that

$$f(\lambda_{\max}(\boldsymbol{L}) \cdot \boldsymbol{I}_d) = \lambda_{\max}(\boldsymbol{L})\|x - y\|^2 - \frac{1}{\lambda_{\max}(\boldsymbol{L})}\|\nabla f(x) - \nabla f(y)\|^2 \geq 0,$$

which yields

$$\|\nabla f(x) - \nabla f(y)\|^2 \leq \lambda^2_{\max}(\boldsymbol{L})\|x - y\|^2 .$$

Since we are working with $\boldsymbol{L} \in \mathbb{S}^d_{++}$, the above inequality implies

$$\|\nabla f(x) - \nabla f(y)\| \leq \lambda_{\max}(\boldsymbol{L})\|x - y\| .$$

This completes the proof.

E Analysis of det-MARINA

We first state three technical lemmas. This first lemma is a slightly different version of [LBZR21, Lemma 2].

Lemma 1. *Assume that function f is $\boldsymbol{L}$ smooth, and $x^{k+1} = x^k - \boldsymbol{D} \cdot g^k$, where $\boldsymbol{D} \in \mathbb{S}^d_{++}$. Then we will have*

$$f(x^{k+1}) \leq f(x^k) - \frac{1}{2}\left\|\nabla f(x^k)\right\|^2_{\boldsymbol{D}} + \frac{1}{2}\left\|g^k - \nabla f(x^k)\right\|^2_{\boldsymbol{D}} - \frac{1}{2}\left\|x^{k+1} - x^k\right\|_{\boldsymbol{D}^{-1}-\boldsymbol{L}}. \tag{32}$$

The following lemma is obtained for any sketch matrix $\boldsymbol{S} \in \mathbb{S}^d_+$ and any two positive definite matrices $\boldsymbol{D}$ and $\boldsymbol{L}$.

Lemma 2. (Property of sketch matrix) *For any sketch matrix $\boldsymbol{S} \in \mathbb{S}^d_+$, a vector $t \in \mathbb{R}^d$, and matrices $\boldsymbol{D}, \boldsymbol{L} \in \mathbb{S}^d_{++}$, we have*

$$\mathbb{E}\left[\left\|\boldsymbol{S}t - t\right\|^2_{\boldsymbol{D}}\right] \leq \lambda_{\max}\left(\boldsymbol{L}^{\frac{1}{2}}\left(\mathbb{E}\left[\boldsymbol{S}\boldsymbol{D}\boldsymbol{S}\right] - \boldsymbol{D}\right)\boldsymbol{L}^{\frac{1}{2}}\right) \cdot \left\|t\right\|^2_{\boldsymbol{L}^{-1}}. \tag{33}$$

E.1 Proof of Theorem 1

Proof of Theorem 1. According to Lemma 1, we have

$$\mathbb{E}\left[f(x^{k+1})\right] \leq \mathbb{E}\left[f(x^k)\right] - \mathbb{E}\left[\frac{1}{2}\left\|\nabla f(x^k)\right\|^2_{\boldsymbol{D}}\right] + \mathbb{E}\left[\frac{1}{2}\left\|g^k - \nabla f(x^k)\right\|^2_{\boldsymbol{D}}\right] - \mathbb{E}\left[\frac{1}{2}\left\|x^{k+1} - x^k\right\|^2_{\boldsymbol{D}^{-1}-\boldsymbol{L}}\right]. \tag{34}$$

We then use the definition of g^{k+1} to derive an upper bound for $\mathbb{E}\left[\left\|g^{k+1} - \nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}}\right]$. Notice that,

$$g^{k+1} = \begin{cases} \nabla f(x^{k+1}) & \text{with probability } p, \\ g^k + \frac{1}{n}\sum_{i=1}^n \boldsymbol{S}^k_i\left(\nabla f_i(x^{k+1}) - \nabla f_i(x^k)\right) & \text{with probability } 1-p. \end{cases}$$

As a result, from the tower property,

$$\begin{aligned}
&\mathbb{E}\left[\left\|g^{k+1} - \nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}} \mid x^{k+1}, x^k\right] \\
&\quad = \mathbb{E}\left[\mathbb{E}\left[\left\|g^{k+1} - \nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}} \mid x^{k+1}, x^k, c_k\right]\right] \\
&\quad = p \cdot \left\|\nabla f(x^{k+1}) - \nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}} \\
&\qquad + (1-p)\cdot \mathbb{E}\left[\left\|g^k + \frac{1}{n}\sum_{i=1}^n \boldsymbol{S}^k_i(\nabla f_i(x^{k+1}) - \nabla f_i(x^k)) - \nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}} \mid x^{k+1}, x^k\right] \\
&\quad = (1-p)\cdot \mathbb{E}\left[\left\|g^k + \frac{1}{n}\sum_{i=1}^n \boldsymbol{S}^k_i(\nabla f_i(x^{k+1}) - \nabla f_i(x^k)) - \nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}} \mid x^{k+1}, x^k\right].
\end{aligned}$$

Using Fact 2, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|g^{k+1} - \nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}} \mid x^{k+1}, x^k\right] \\
&\quad = (1-p)\cdot \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{S}^k_i(\nabla f_i(x^{k+1}) - \nabla f_i(x^k)) - \left(\nabla f(x^{k+1}) - \nabla f(x^k)\right)\right\|^2_{\boldsymbol{D}} \mid x^{k+1}, x^k\right] \\
&\qquad + (1-p)\cdot \left\|g^k - \nabla f(x^k)\right\|^2_{\boldsymbol{D}} \\
&\quad = (1-p)\cdot \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \left(\boldsymbol{S}^k_i(\nabla f_i(x^{k+1}) - \nabla f_i(x^k)) - (\nabla f_i(x^{k+1}) - \nabla f_i(x^k))\right)\right\|^2_{\boldsymbol{D}} \mid x^{k+1}, x^k\right] \\
&\qquad + (1-p)\cdot \left\|g^k - \nabla f(x^k)\right\|^2_{\boldsymbol{D}}.
\end{aligned}$$

Notice that the sketch matrix is unbiased, thus we have

$$\mathbb{E}\left[\boldsymbol{S}_i^k\left(\nabla f_i(x^{k+1})-\nabla f_i(x^k)\right)\mid x^{k+1},x^k\right]=\nabla f_i(x^{k+1})-\nabla f_i(x^k),$$

and any two random vectors in the set $\{\boldsymbol{S}_i^k(\nabla f_i(x^{k+1})-\nabla f_i(x^k))\}_{i=1}^n$ are independent from each other, if x^{k+1} and x^k are fixed. Therefore, we have

$$\begin{aligned}&\mathbb{E}\left[\left\|g^{k+1}-\nabla f(x^{k+1})\right\|_{\boldsymbol{D}}^2\mid x^{k+1},x^k\right]\\&\quad=\frac{1-p}{n^2}\sum_{i=1}^n\mathbb{E}\left[\left\|\boldsymbol{S}_i^k(\nabla f_i(x^{k+1})-\nabla f_i(x^k))-(\nabla f_i(x^{k+1})-\nabla f_i(x^k))\right\|_{\boldsymbol{D}}^2\mid x^{k+1},x^k\right]\\&\qquad+(1-p)\cdot\left\|g^k-\nabla f(x^k)\right\|_{\boldsymbol{D}}^2.\end{aligned}\tag{35}$$

Lemma 2 yields

$$\begin{aligned}&\mathbb{E}\left[\left\|\boldsymbol{S}_i^k(\nabla f_i(x^{k+1})-\nabla f_i(x^k))-(\nabla f_i(x^{k+1})-\nabla f_i(x^k))\right\|_{\boldsymbol{D}}^2\mid x^{k+1},x^k\right]\\&\qquad\leq\lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{D}\boldsymbol{S}_i^k\right]-\boldsymbol{D}\right)\boldsymbol{L}_i^{\frac{1}{2}}\right)\left\|\nabla f_i(x^{k+1})-\nabla f_i(x^k)\right\|_{\boldsymbol{L}_i^{-1}}^2.\end{aligned}\tag{36}$$

Assumption 3 implies

$$\begin{aligned}&\mathbb{E}\left[\left\|\boldsymbol{S}_i^k(\nabla f_i(x^{k+1})-\nabla f_i(x^k))-(\nabla f_i(x^{k+1})-\nabla f_i(x^k))\right\|_{\boldsymbol{D}}^2\mid x^{k+1},x^k\right]\\&\qquad\leq\lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{D}\boldsymbol{S}_i^k\right]-\boldsymbol{D}\right)\boldsymbol{L}_i^{\frac{1}{2}}\right)\left\|x^{k+1}-x^k\right\|_{\boldsymbol{L}_i}^2.\end{aligned}\tag{37}$$

Plugging back (37) into (35), we deduce

$$\begin{aligned}&\mathbb{E}\left[\left\|g^{k+1}-\nabla f(x^{k+1})\right\|_{\boldsymbol{D}}^2\mid x^{k+1},x^k\right]\\&\quad\leq\frac{1-p}{n^2}\sum_{i=1}^n\lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{D}\boldsymbol{S}_i^k\right]-\boldsymbol{D}\right)\boldsymbol{L}_i^{\frac{1}{2}}\right)\left\|x^{k+1}-x^k\right\|_{\boldsymbol{L}_i}^2+(1-p)\cdot\left\|g^k-\nabla f(x^k)\right\|_{\boldsymbol{D}}^2.\end{aligned}$$

Replacing $\boldsymbol{L}_i^{-1}$ with $\boldsymbol{L}^{-1/2}\boldsymbol{L}^{1/2}\boldsymbol{L}_i^{-1}\boldsymbol{L}^{1/2}\boldsymbol{L}^{-1/2}$, we then rewrite the $\boldsymbol{L}_i$-norm in the first term of RHS by the $\boldsymbol{L}$-norm:

$$\begin{aligned}&\mathbb{E}\left[\left\|g^{k+1}-\nabla f(x^{k+1})\right\|_{\boldsymbol{D}}^2\mid x^{k+1},x^k\right]\\&\quad=\frac{1-p}{n^2}\sum_{i=1}^n\lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{D}\boldsymbol{S}_i^k\right]-\boldsymbol{D}\right)\boldsymbol{L}_i^{\frac{1}{2}}\right)\left(\boldsymbol{L}^{\frac{1}{2}}(x^{k+1}-x^k)\right)^\top\boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{L}_i\boldsymbol{L}^{-\frac{1}{2}}\left(\boldsymbol{L}^{\frac{1}{2}}(x^{k+1}-x^k)\right)\\&\qquad+(1-p)\cdot\left\|g^k-\nabla f(x^k)\right\|_{\boldsymbol{D}}^2\\&\quad\leq\frac{1-p}{n^2}\sum_{i=1}^n\lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{D}\boldsymbol{S}_i^k\right]-\boldsymbol{D}\right)\boldsymbol{L}_i^{\frac{1}{2}}\right)\cdot\lambda_{\max}\left(\boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{L}_i\boldsymbol{L}^{-\frac{1}{2}}\right)\left\|x^{k+1}-x^k\right\|_{\boldsymbol{L}}^2\\&\qquad+(1-p)\cdot\left\|g^k-\nabla f(x^k)\right\|_{\boldsymbol{D}}^2.\end{aligned}$$

We further use Fact 5 to upper bound $\lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{D}\boldsymbol{S}_i^k\right]-\boldsymbol{D}\right)\boldsymbol{L}_i^{\frac{1}{2}}\right)$ by the product of $\lambda_{\max}\left(\boldsymbol{L}_i\right)$ and $\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{D}\boldsymbol{S}_i^k\right]-\boldsymbol{D}\right)$. This allows us to simplify the expression since $\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{D}\boldsymbol{S}_i^k\right]-\boldsymbol{D}\right)$ is independent of the index i. Notice that we have already defined

$$R(\boldsymbol{D},\mathcal{S})=\frac{1}{n}\sum_{i=1}^n\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{D}\boldsymbol{S}_i^k\right]-\boldsymbol{D}\right)\cdot\lambda_{\max}\left(\boldsymbol{L}_i\right)\cdot\lambda_{\max}\left(\boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{L}_i\boldsymbol{L}^{-\frac{1}{2}}\right).$$

Taking expectation, using tower property and using the definition above, we deduce

$$\mathbb{E}\left[\left\|g^{k+1}-\nabla f(x^{k+1})\right\|_{\boldsymbol{D}}^2\right]\leq\frac{(1-p)\cdot R(\boldsymbol{D},\mathcal{S})}{n}\mathbb{E}\left[\left\|x^{k+1}-x^k\right\|_{\boldsymbol{L}}^2\right]+(1-p)\mathbb{E}\left[\left\|g^k-\nabla f(x^k)\right\|_{\boldsymbol{D}}^2\right].\tag{38}$$

We construct the Lyapunov function Φ_k as follows,

$$\Phi_k = f(x^k) - f^\star + \frac{1}{2p}\left\|g^k - \nabla f(x^k)\right\|_{\boldsymbol{D}}^2. \tag{39}$$

Using (34) and (38), we are able to get

$$\begin{aligned}
\mathbb{E}\left[\Phi_{k+1}\right] &\leq \mathbb{E}\left[f(x^k) - f^\star\right] - \frac{1}{2}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|_{\boldsymbol{D}}^2\right] + \frac{1}{2}\mathbb{E}\left[\left\|g^k - \nabla f(x^k)\right\|_{\boldsymbol{D}}^2\right] - \frac{1}{2}\mathbb{E}\left[\left\|x^{k+1} - x^k\right\|_{\boldsymbol{D}^{-1}-\boldsymbol{L}}^2\right] \\
&\quad + \frac{1}{2p}\left[\frac{(1-p)\cdot R(\boldsymbol{D},\mathcal{S})}{n}\mathbb{E}\left[\left\|x^{k+1} - x^k\right\|_{\boldsymbol{L}}^2\right] + (1-p)\cdot\mathbb{E}\left[\left\|g^k - \nabla f(x^k)\right\|_{\boldsymbol{D}}^2\right]\right] \\
&= \mathbb{E}\left[\Phi_k\right] - \frac{1}{2}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|_{\boldsymbol{D}}^2\right] \\
&\quad + \left(\frac{(1-p)\cdot R(\boldsymbol{D},\mathcal{S})}{2np}\mathbb{E}\left[\left\|x^{k+1} - x^k\right\|_{\boldsymbol{L}}^2\right] - \frac{1}{2}\mathbb{E}\left[\left\|x^{k+1} - x^k\right\|_{\boldsymbol{D}^{-1}-\boldsymbol{L}}^2\right]\right) \\
&= \mathbb{E}\left[\Phi_k\right] - \frac{1}{2}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|_{\boldsymbol{D}}^2\right] \\
&\quad + \frac{1}{2}\left(\frac{(1-p)\cdot R(\boldsymbol{D},\mathcal{S})}{np}\mathbb{E}\left[\left\|x^{k+1} - x^k\right\|_{\boldsymbol{L}}^2\right] - \mathbb{E}\left[\left\|x^{k+1} - x^k\right\|_{\boldsymbol{D}^{-1}-\boldsymbol{L}}^2\right]\right).
\end{aligned}$$

We can rewrite the last term as

$$\mathbb{E}\left[(x^{k+1} - x^k)^\top\left[\frac{(1-p)\cdot R(\boldsymbol{D},\mathcal{S})}{np}\boldsymbol{L} + \boldsymbol{L} - \boldsymbol{D}^{-1}\right](x^{k+1} - x^k)\right]. \tag{40}$$

We require the matrix in between to be negative semi-definite, which is

$$\boldsymbol{D}^{-1} \succeq \left(\frac{(1-p)\cdot R(\boldsymbol{D},\mathcal{S})}{np} + 1\right)\boldsymbol{L}.$$

This leads to the result that the expression (40) is always non-positive. After dropping the last term, the relation between $\mathbb{E}\left[\Phi_{k+1}\right]$ and $\mathbb{E}\left[\Phi_k\right]$ becomes

$$\mathbb{E}\left[\Phi_{k+1}\right] \leq \mathbb{E}\left[\Phi_k\right] - \frac{1}{2}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|_{\boldsymbol{D}}^2\right]. \tag{41}$$

Unrolling this recurrence, we get

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|_{\boldsymbol{D}}^2\right] \leq \frac{2\left(\mathbb{E}\left[\Phi_0\right] - \mathbb{E}\left[\Phi_K\right]\right)}{K}. \tag{42}$$

The left hand side can viewed as $\mathbb{E}\left[\left\|\nabla f(\tilde{x}^K)\right\|_{\boldsymbol{D}}^2\right]$, where $\tilde{x}^K$ is drawn uniformly at random from $\{x_k\}_{k=0}^{K-1}$. From $\Phi_K > 0$, we obtain

$$\begin{aligned}
\frac{2\left(\mathbb{E}\left[\Phi_0\right] - \mathbb{E}\left[\Phi_K\right]\right)}{K} &\leq \frac{2\Phi_0}{K} \\
&= \frac{2\left(f(x^0) - f^\star + \frac{1}{2p}\left\|g^0 - \nabla f(x^0)\right\|_{\boldsymbol{D}}^2\right)}{K} \\
&= \frac{2\left(f(x^0) - f^\star\right)}{K}.
\end{aligned}$$

Plugging in the simplified result into (42), and performing determinant normalization, we get

$$\mathbb{E}\left[\left\|\nabla f(\tilde{x}^K)\right\|_{\frac{\boldsymbol{D}}{\det(\boldsymbol{D})^{1/d}}}^2\right] \leq \frac{2\left(f(x^0) - f^\star\right)}{\det(\boldsymbol{D})^{1/d}K}. \tag{43}$$

This completes the proof. □

A remark on the convergence result obtained in the proof of Theorem 1 is given here,

Remark 7. *We can achieve a slightly more refined stepsize condition than (7) for* `det-MARINA`*, which is given as follows*

$$\boldsymbol{D} \succeq \left(\frac{(1-p)\cdot \tilde{R}(\boldsymbol{D},\mathcal{S})}{np} + 1 \right) \boldsymbol{L}, \tag{44}$$

where

$$\tilde{R}(\boldsymbol{D},\mathcal{S}) := \frac{1}{n}\sum_{i=1}^{n} \lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}} \left(\mathbb{E}\left[\boldsymbol{S}_i^k \boldsymbol{D} \boldsymbol{S}_i^k \right] - \boldsymbol{D} \right) \boldsymbol{L}_i^{\frac{1}{2}} \right) \cdot \lambda_{\max}\left(\boldsymbol{L}^{-\frac{1}{2}} \boldsymbol{L}_i \boldsymbol{L}^{-\frac{1}{2}} \right).$$

This is obtained if we do not use Fact 5 to upper bound $\lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}} \left(\mathbb{E}\left[\boldsymbol{S}_i^k \boldsymbol{D} \boldsymbol{S}_i^k \right] - \boldsymbol{D} \right) \boldsymbol{L}_i^{\frac{1}{2}} \right)$ *by the product of* $\lambda_{\max}\left(\boldsymbol{L}_i\right)$ *and* $\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k \boldsymbol{D} \boldsymbol{S}_i^k \right] - \boldsymbol{D} \right)$*. However, (44) results in a condition that is much harder to solve even if we assume* $\boldsymbol{D} = \gamma \cdot \boldsymbol{W}$*. So instead of using the more refined condition (44), we turn to (7). Notice that both of the two conditions (44) and (7) reduce to the stepsize condition for* `MARINA` *in the scalar setting.*

E.2 Comparison of Different Stepsizes

In Corollary 2, we focus on the special stepsize where we fix $\boldsymbol{W} = \boldsymbol{L}^{-1}$, and show that in this case `det-MARINA` always beats `MARINA` in terms of both iteration and communication complexities. However, other choices for $\boldsymbol{W}$ are also possible. Specifically, we consider the cases where $\boldsymbol{W} = \operatorname{diag}^{-1}(\boldsymbol{L})$ and $\boldsymbol{W} = \boldsymbol{I}_d$.

E.2.1 The Diagonal Case

We consider $\boldsymbol{W} = \operatorname{diag}^{-1}(\boldsymbol{L})$. The following corollary describes the optimal stepsize and the iteration complexity.

Corollary 4. *If we take* $\boldsymbol{W} = \operatorname{diag}^{-1}(\boldsymbol{L})$ *in Corollary 1, then the optimal stepsize satisfies*

$$\boldsymbol{D}^*_{\operatorname{diag}^{-1}(\boldsymbol{L})} = \frac{2}{1+\sqrt{1+4\alpha\beta\cdot\Lambda_{\operatorname{diag}^{-1}(\boldsymbol{L}),\mathcal{S}}}} \cdot \operatorname{diag}^{-1}(\boldsymbol{L}). \tag{45}$$

This stepsize results in a better iteration complexity of `det-MARINA` *compared to scalar* `MARINA`*.*

From this corollary we know that `det-MARINA` has a better iteration complexity when $\boldsymbol{W} = \operatorname{diag}^{-1}(\boldsymbol{L})$. And since the same sketch is used for `MARINA` and `det-MARINA`, the communication complexity is improved as well. However, in general there is no clear relation between the iteration complexity of $\boldsymbol{W} = \boldsymbol{L}^{-1}$ case and $\boldsymbol{W} = \operatorname{diag}^{-1}(\boldsymbol{L})$ case. This is also confirmed by one of our experiments, see Figure 4 to see the comparison of `det-MARINA` using optimal stepsizes in different cases.

E.2.2 The Identity Case

In this setting, $\boldsymbol{W}$ is the d-dimensional identity matrix $\boldsymbol{I}_d$. Then the stepsize of our algorithm reduces to a scalar γ, where γ is determined through Corollary 1. Notice that in this case we do not reduce to the standard `MARINA` case because we are still using the matrix Lipschitz gradient assumption with $\boldsymbol{L} \in \mathbb{S}^d_{++}$.

Corollary 5. *If we take* $\boldsymbol{W} = \boldsymbol{I}_d$*, the optimal stepsize is given by*

$$\boldsymbol{D}^*_{\boldsymbol{I}_d} = \frac{2}{1+\sqrt{1+4\alpha\beta\frac{1}{\lambda_{\max}(\boldsymbol{L})}\cdot\omega}} \cdot \frac{\boldsymbol{I}_d}{\lambda_{\max}(\boldsymbol{L})}. \tag{46}$$

This stepsize results in a better iteration complexity of `det-MARINA` *compared to scalar* `MARINA`*.*

The result in this corollary tells us that using scalar stepsize with matrix Lipschitz gradient assumption alone can result in acceleration of `MARINA`. However, the use of matrix stepsize allows us to also take into consideration the "structure" of the stepsize, thus allows more flexibility. When the structure of the stepsize is chosen properly, combining matrix gradient Lipschitzness and matrix stepsize can result in a faster rate, as it can also be observed from the experiments in Figure 4. The choices of $\boldsymbol{W}$ we consider here are in some sense inspired by the matrix stepsize `GD`, where the optimal stepsize is $\boldsymbol{L}^{-1}$. In general, how to identify the best structure for the matrix stepsize remains a open problem.

E.3 Proofs of the Corollaries

E.3.1 Proof of Corollary 1

We start with rewriting condition (7) as

$$\left(\frac{1-p}{np}\cdot R\left(\boldsymbol{D},\mathcal{S}\right)+1\right)\boldsymbol{D}^{\frac{1}{2}}\boldsymbol{L}\boldsymbol{D}^{\frac{1}{2}}\preceq\boldsymbol{I}_d.$$

Plugging in the definition of $R(\boldsymbol{D},\mathcal{S})$ and $\boldsymbol{D}=\gamma\boldsymbol{W}$, we get

$$\gamma\left(\frac{1-p}{np}\cdot\frac{1}{n}\sum_{i=1}^{n}\lambda_{\max}\left(\boldsymbol{L}_i\right)\lambda_{\max}\left(\boldsymbol{L}^{-1}\boldsymbol{L}_i\right)\cdot\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{W}\boldsymbol{S}_i^k\right]-\boldsymbol{W}\right)\cdot\gamma+1\right)\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{L}\boldsymbol{W}^{\frac{1}{2}}\preceq\boldsymbol{I}_d.$$

This generalized inequality is equivalent to the following inequality,

$$\gamma\left(\frac{1-p}{np}\cdot\frac{1}{n}\sum_{i=1}^{n}\lambda_{\max}\left(\boldsymbol{L}_i\right)\lambda_{\max}\left(\boldsymbol{L}^{-1}\boldsymbol{L}_i\right)\cdot\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{W}\boldsymbol{S}_i^k\right]-\boldsymbol{W}\right)\cdot\gamma+1\right)\cdot\lambda_{\max}\left(\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{L}\boldsymbol{W}^{\frac{1}{2}}\right)\leq 1,$$

which is a quadratic inequality on γ. Notice that we have already defined

$$\alpha=\frac{1-p}{np};\qquad\beta=\frac{1}{n}\sum_{i=1}^{n}\lambda_{\max}\left(\boldsymbol{L}_i\right)\cdot\lambda_{\max}\left(\boldsymbol{L}^{-1}\boldsymbol{L}_i\right);$$
$$\Lambda_{\boldsymbol{W},\mathcal{S}}=\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{W}\boldsymbol{S}_i^k\right]-\boldsymbol{W}\right);\qquad\lambda_{\boldsymbol{W}}=\lambda_{\max}^{-1}\left(\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{L}\boldsymbol{W}^{\frac{1}{2}}\right).$$

As a result, the above inequality can be written equivalently as

$$\alpha\beta\Lambda_{\boldsymbol{W},\mathcal{S}}\cdot\gamma^2+\gamma-\lambda_{\boldsymbol{W}}\leq 0,$$

which yields the upper bound on γ

$$\gamma\leq\frac{\sqrt{1+4\alpha\beta\cdot\Lambda_{\boldsymbol{W},\mathcal{S}}\lambda_{\boldsymbol{W}}}-1}{2\alpha\beta\cdot\Lambda_{\boldsymbol{W},\mathcal{S}}}.$$

Since $\sqrt{1+4\alpha\beta\cdot\Lambda_{\boldsymbol{W},\mathcal{S}}\lambda_{\boldsymbol{W}}}+1>0$, we can simplify the result as

$$\gamma\leq\frac{2\lambda_{\boldsymbol{W}}}{1+\sqrt{1+4\alpha\beta\cdot\Lambda_{\boldsymbol{W},\mathcal{S}}\lambda_{\boldsymbol{W}}}}.$$

E.3.2 Proof of Corollary 2

The equation (15) directly follows from plugging $\boldsymbol{W}=\boldsymbol{L}^{-1}$ into (13). The optimal stepsize is obtained as the product of γ and $\boldsymbol{L}^{-1}$. The iteration complexity of MARINA, according to [GBLR21], is

$$K\geq K_1=\mathcal{O}\left(\frac{\Delta_0 L}{\varepsilon^2}\left(1+\sqrt{\frac{(1-p)\omega}{pn}}\right)\right).\tag{47}$$

On the other hand,

$$\det(\boldsymbol{L})^{\frac{1}{d}}\leq\lambda_{\max}\left(\boldsymbol{L}\right)=L.\tag{48}$$

In addition, using the inequality

$$\sqrt{1+4t}\leq 1+2\sqrt{t},\tag{49}$$

which holds for any $t\geq 0$, we have the following bound

$$\frac{\left(1+\sqrt{1+4\alpha\beta\cdot\Lambda_{\boldsymbol{L}^{-1},\mathcal{S}}}\right)}{2}\leq 1+\sqrt{\alpha\beta\cdot\Lambda_{\boldsymbol{L}^{-1},\mathcal{S}}}.$$

Next we prove that

$$1+\sqrt{\alpha\beta\cdot\Lambda_{\boldsymbol{L}^{-1},\mathcal{S}}}\leq 1+\sqrt{\frac{(1-p)}{pn}\cdot\omega},\tag{50}$$

which is equivalent to proving

$$\frac{1}{n}\sum_{i=1}^{n}\lambda_{\max}\left(\boldsymbol{L}_i\right)\lambda_{\max}\left(\boldsymbol{L}_i\boldsymbol{L}^{-1}\right)\cdot\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{L}^{-1}\boldsymbol{S}_i^k\right]-\boldsymbol{L}^{-1}\right)\leq\omega.$$

The left hand side can be upper bounded by,

$$\frac{1}{n}\sum_{i=1}^{n}\lambda_{\max}\left(\boldsymbol{L}_i\right)\lambda_{\max}\left(\boldsymbol{L}^{-1}\boldsymbol{L}_i\right)\cdot\lambda_{\max}\left(\boldsymbol{L}^{-1}\right)\cdot\frac{\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{L}^{-1}\boldsymbol{S}_i^k\right]-\boldsymbol{L}^{-1}\right)}{\lambda_{\max}\left(\boldsymbol{L}^{-1}\right)}\leq\frac{\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{L}^{-1}\boldsymbol{S}_i^k\right]-\boldsymbol{L}^{-1}\right)}{\lambda_{\max}\left(\boldsymbol{L}^{-1}\right)},$$

where the inequality is a consequence of Proposition 2. We further bound the last term with

$$\begin{aligned}\frac{\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{L}^{-1}\boldsymbol{S}_i^k\right]-\boldsymbol{L}^{-1}\right)}{\lambda_{\max}\left(\boldsymbol{L}^{-1}\right)} &= \lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\cdot\frac{\boldsymbol{L}^{-1}}{\lambda_{\max}(\boldsymbol{L}^{-1})}\cdot\boldsymbol{S}_i^k\right]-\frac{\boldsymbol{L}^{-1}}{\lambda_{\max}\left(\boldsymbol{L}^{-1}\right)}\right)\\ &\leq \lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{S}_i^k\right]-\boldsymbol{I}_d\right)\\ &=: \omega.\end{aligned}$$

Here, the last inequality is due to the monotone behavior of the mapping $\boldsymbol{X}\mapsto\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{X}\boldsymbol{S}_i^k\right]-\boldsymbol{X}\right)$ with $\boldsymbol{X}\in\mathbb{S}_{++}^d$, which can be shown as follows, let us pick any $\boldsymbol{X}_1,\boldsymbol{X}_2\in\mathbb{S}_{++}^d$ and $\boldsymbol{X}_1\preceq\boldsymbol{X}_2$,

$$\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{X}_2\boldsymbol{S}_i^k\right]-\boldsymbol{X}_2\right)-\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{X}_1\boldsymbol{S}_i^k\right]-\boldsymbol{X}_1\right)=\mathbb{E}\left[\boldsymbol{S}_i^k\left(\boldsymbol{X}_2-\boldsymbol{X}_1\right)\boldsymbol{S}_i^k\right]-\left(\boldsymbol{X}_2-\boldsymbol{X}_1\right)\succeq\boldsymbol{O}_d.$$

The above inequality is due to the convexity of the mapping $\boldsymbol{S}_i^k\mapsto\boldsymbol{S}_i^k\boldsymbol{X}\boldsymbol{S}_i^k$. As a result, we have

$$\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{X}_2\boldsymbol{S}_i^k\right]-\boldsymbol{X}_2\right)\geq\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{X}_1\boldsymbol{S}_i^k\right]-\boldsymbol{X}_1\right),$$

whenever $\boldsymbol{X}_2\succeq\boldsymbol{X}_1$. Due to the fact that

$$\frac{\boldsymbol{L}^{-1}}{\lambda_{\max}\left(\boldsymbol{L}^{-1}\right)}\preceq\boldsymbol{I}_d,$$

we have

$$\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\cdot\frac{\boldsymbol{L}^{-1}}{\lambda_{\max}(\boldsymbol{L}^{-1})}\cdot\boldsymbol{S}_i^k\right]-\frac{\boldsymbol{L}^{-1}}{\lambda_{\max}\left(\boldsymbol{L}^{-1}\right)}\right)\leq\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\cdot\boldsymbol{I}_d\cdot\boldsymbol{S}_i^k\right]-\boldsymbol{I}_d\right)=\omega.$$

Combining (48) and (50), we know that the iteration complexity of `det-MARINA` is always better than that of `MARINA`.

E.3.3 Proof of Corollary 3

Since we run the algorithm for K iterations, the number of bits sent in expectation should be

$$\mathcal{O}(d+K(pd+(1-p)\zeta_{\mathcal{S}}))=\mathcal{O}((Kp+1)d+(1-p)K\zeta_{\mathcal{S}}).$$

The special case where we choose $p=\zeta_{\mathcal{S}}/d$ indicates that

$$\alpha=\frac{1-p}{np}=\frac{1}{n}\left(\frac{d}{\zeta_{\mathcal{S}}}-1\right).$$

In order to reach an error of ε^2, we need

$$K=\mathcal{O}\left(\frac{\Delta_0\cdot\det(\boldsymbol{L})^{\frac{1}{d}}}{\varepsilon^2}\cdot\left(1+\sqrt{1+\frac{4\beta}{n}\left(\frac{d}{\zeta_{\mathcal{S}}}-1\right)\cdot\Lambda_{\boldsymbol{L}^{-1},\mathcal{S}}}\right)\right),$$

which is the iteration complexity. Applying once again (49) and using the fact that $p=\zeta_{\mathcal{S}}/d$, the communication complexity in this case is given by

$$\begin{aligned}&\mathcal{O}\left(d+\frac{\Delta_0\cdot\det(\boldsymbol{L})^{\frac{1}{d}}}{\varepsilon^2}\cdot\left(1+\sqrt{1+\frac{4\beta}{n}\left(\frac{d}{\zeta_{\mathcal{S}}}-1\right)\cdot\Lambda_{\boldsymbol{L}^{-1},\mathcal{S}}}\right)\cdot(pd+(1-p)\zeta_{\mathcal{S}})\right)\\ &\leq\mathcal{O}\left(d+\frac{2\Delta_0\cdot\det(\boldsymbol{L})^{\frac{1}{d}}}{\varepsilon^2}\cdot\left(1+\sqrt{\frac{\beta}{n}\left(\frac{d}{\zeta_{\mathcal{S}}}-1\right)\cdot\Lambda_{\boldsymbol{L}^{-1},\mathcal{S}}}\right)\cdot(pd+(1-p)\zeta_{\mathcal{S}})\right)\\ &\leq\mathcal{O}\left(d+\frac{4\Delta_0\cdot\det(\boldsymbol{L})^{\frac{1}{d}}}{\varepsilon^2}\cdot\left(\zeta_{\mathcal{S}}+\sqrt{\frac{\beta\cdot\Lambda_{\boldsymbol{L}^{-1},\mathcal{S}}}{n}\cdot\zeta_{\mathcal{S}}(d-\zeta_{\mathcal{S}})}\right)\right).\end{aligned}$$

Ignoring the coefficient we get

$$\mathcal{O}\left(d+\frac{\Delta_0\cdot\det(\boldsymbol{L})^{\frac{1}{d}}}{\varepsilon^2}\cdot\left(\zeta_{\mathcal{S}}+\sqrt{\frac{\beta\cdot\Lambda_{\boldsymbol{L}^{-1},\mathcal{S}}}{n}\cdot\zeta_{\mathcal{S}}(d-\zeta_{\mathcal{S}})}\right)\right).$$

This completes the proof.

E.3.4 Proof of Corollary 4

The proof is similar to the proof of Corollary 2. Applying Corollary 1, notice that in this case

$$\lambda_{\operatorname{diag}^{-1}(\boldsymbol{L})}=\lambda_{\max}^{-1}\left(\operatorname{diag}^{-\frac{1}{2}}(\boldsymbol{L})\,\boldsymbol{L}\operatorname{diag}^{-\frac{1}{2}}(\boldsymbol{L})\right)=1,$$

we can obtain $\boldsymbol{D}^*_{\operatorname{diag}^{-1}(\boldsymbol{L})}$. The iteration complexity is thus given by

$$\mathcal{O}\left(\frac{\det\left(\operatorname{diag}(\boldsymbol{L})\right)^{\frac{1}{d}}\cdot\Delta_0}{\varepsilon^2}\cdot\left(\frac{1+\sqrt{1+4\alpha\beta\Lambda_{\operatorname{diag}^{-1}(\boldsymbol{L}),\mathcal{S}}}}{2}\right)\right).$$

We now compare it to the iteration complexity of `MARINA`, which is given in (47). We know that each diagonal element $\boldsymbol{L}_{jj}$ satisfies $\boldsymbol{L}_{jj}\leq\lambda_{\max}(\boldsymbol{L})=L$ for $j=1,\ldots,d$. As a result,

$$\det\left(\operatorname{diag}(\boldsymbol{L})\right)^{\frac{1}{d}}\leq L. \tag{51}$$

From (49), we deduce

$$\frac{1+\sqrt{1+4\alpha\beta\cdot\Lambda_{\operatorname{diag}^{-1}(\boldsymbol{L}),\mathcal{S}}}}{2}\leq 1+\sqrt{\alpha\beta\cdot\Lambda_{\operatorname{diag}^{-1}(\boldsymbol{L}),\mathcal{S}}}.$$

Now, let us prove the below inequality

$$1+\sqrt{\alpha\beta\cdot\Lambda_{\operatorname{diag}^{-1}(\boldsymbol{L}),\mathcal{S}}}\leq 1+\sqrt{\frac{(1-p)}{pn}\cdot\omega}. \tag{52}$$

The latter is equivalent to

$$\beta\cdot\Lambda_{\operatorname{diag}^{-1}(\boldsymbol{L}),\mathcal{S}}\leq\omega.$$

Plugging in the definition of β,ω and $\Lambda_{\operatorname{diag}^{-1}(\boldsymbol{L}),\mathcal{S}}$ and using the relation given in Proposition 2, we obtain,

$$\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\frac{\operatorname{diag}^{-1}(\boldsymbol{L})}{\lambda_{\max}(\boldsymbol{L}^{-1})}\boldsymbol{S}_i^k-\frac{\operatorname{diag}^{-1}(\boldsymbol{L})}{\lambda_{\max}(\boldsymbol{L}^{-1})}\right]\right)\leq\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k\boldsymbol{I}_d\boldsymbol{S}_i^k\right]-\boldsymbol{I}_d\right).$$

Thus, it is enough to prove that

$$\frac{\operatorname{diag}^{-1}(\boldsymbol{L})}{\lambda_{\max}(\boldsymbol{L}^{-1})}\preceq\boldsymbol{I}_d.$$

We can further simplify the above inequality as

$$\lambda_{\min}(\boldsymbol{L})\leq\lambda_{\min}\left(\operatorname{diag}(\boldsymbol{L})\right),$$

which is always true for any $\boldsymbol{L}\in\mathbb{S}^d_{++}$. Combining (51) and (52) we conclude the proof.

E.3.5 Proof of Corollary 5

Using the explicit formula for the optimal stepsize $\boldsymbol{D}^*_{\boldsymbol{I}_d}$, we deduce the following iteration complexity for

$$\mathcal{O}\left(\frac{\lambda_{\max}(\boldsymbol{L})\,\Delta_0}{\varepsilon^2}\cdot\left(\frac{1+\sqrt{1+4\alpha\beta\frac{\omega}{\lambda_{\max}(\boldsymbol{L})}}}{2}\right)\right). \tag{53}$$

We first recall

$$\lambda_{\max}(\boldsymbol{L})=L. \tag{54}$$

Using (49) we obtain

$$\frac{1+\sqrt{1+4\alpha\beta\frac{\omega}{\lambda_{\max}(\boldsymbol{L})}}}{2} \leq 1+\sqrt{\alpha\beta\frac{\omega}{\lambda_{\max}\left(\boldsymbol{L}\right)}}.$$

The comparison of two iteration complexities, given in (53) and (47) reduces to

$$1+\sqrt{\alpha\beta\frac{\omega}{\lambda_{\max}\left(\boldsymbol{L}\right)}} \leq 1+\sqrt{\frac{1-p}{np}\omega}. \tag{55}$$

This is equivalent to

$$\beta\cdot\frac{1}{\lambda_{\max}\left(\boldsymbol{L}\right)} \leq 1.$$

Utilizing Proposition 2, the above inequality can be rewritten as

$$\frac{1}{\lambda_{\max}\left(\boldsymbol{L}^{-1}\right)\cdot\lambda_{\max}\left(\boldsymbol{L}\right)} \leq 1,$$

which is exactly

$$\lambda_{\min}\left(\boldsymbol{L}\right) \leq \lambda_{\max}\left(\boldsymbol{L}\right).$$

This concludes the proof.

F Distributed `det-CGD`

This section a is brief summary of the distributed `det-CGD` algorithm and its theoretical analysis. The details can be found in [LKR23].

The algorithm follows the standard FL paradigm. See the pseudocode in Algorithm 2.

Algorithm 2 Distributed `det-CGD`

Input: Starting point x^0, stepsize matrix $\boldsymbol{D}$, number of iterations K
for $k=0,1,2,\ldots,K-1$ **do**
  The devices in parallel:
  sample $\boldsymbol{S}_i^k \sim \mathcal{S}$;
  compute $\boldsymbol{S}_i^k\nabla f_i(x^k)$;
  broadcast $\boldsymbol{S}_i^k\nabla f_i(x^k)$.
  The server:
  combines $g^k=\frac{1}{n}\sum_{i=1}^n \boldsymbol{S}_i^k\nabla f_i(x^k)$;
  computes $x^{k+1}=x^k-\boldsymbol{D}g^k$;
  broadcasts x^{k+1}.
end for
Return: x^K

Below is the main convergence result for the algorithm. Its proof can be found in [LKR23]

Theorem 2. *Suppose that f is $\boldsymbol{L}$-smooth. Under the Assumptions 1,2, if the stepsize satisfies*

$$\boldsymbol{DLD} \preceq \boldsymbol{D}, \tag{56}$$

then the following convergence bound is true for the iteration of Algorithm 2:

$$\min_{0\leq k\leq K-1}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2_{\frac{\boldsymbol{D}}{\det(\boldsymbol{D})^{1/d}}}\right] \leq \frac{2(1+\frac{\lambda_{\boldsymbol{D}}}{n})^K\left(f(x^0)-f^\star\right)}{\det(\boldsymbol{D})^{1/d}\,K}+\frac{2\lambda_{\boldsymbol{D}}\Delta^\star}{\det(\boldsymbol{D})^{1/d}\,n}, \tag{57}$$

where $\Delta^\star := f^\star-\frac{1}{n}\sum_{i=1}^n f_i^\star$ and

$$\lambda_{\boldsymbol{D}} := \max_i\left\{\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{L}_i^{\frac{1}{2}}\left(\boldsymbol{S}_i^k-\boldsymbol{I}_d\right)\boldsymbol{DLD}\left(\boldsymbol{S}_i^k-\boldsymbol{I}_d\right)\boldsymbol{L}_i^{\frac{1}{2}}\right]\right)\right\}.$$

Remark 8. *On the right hand side of (57) we observe that increasing K will only reduce the first term, that corresponds to the convergence error. Whereas, the second term, which does not depend on K, will remain constant, if the other parameters of the algorithm are fixed. This testifies to the neighborhood phenomenon which we discussed in Appendix B.*

Remark 9. *If the stepsize satisfies the below conditions,*

$$\boldsymbol{DLD} \preceq \boldsymbol{D}, \quad \lambda_{\boldsymbol{D}} \leq \min\left\{\frac{n}{K}, \frac{n\varepsilon^2}{4\Delta^\star}\det(\boldsymbol{D})^{1/d}\right\}, \quad K \geq \frac{12(f(x^0)-f^\star)}{\det(\boldsymbol{D})^{1/d}\,\varepsilon^2}, \tag{58}$$

then we obtain ε-stationary point.

One can see that in the convergence guarantee of `det-CGD` in the distributed case, the result (57) is non variance reduced. Because of this limitation, in order to reach a ε stationary point, the stepsize condition in (58) is restrictive.

G Extension of `det-CGD2` in `MARINA` Form

In this section we want to extend `det-CGD2` into its variance reduced counterpart in `MARINA` form.

Algorithm 3 det-CGD2-VR

Input: starting point x^0, stepsize matrix $\boldsymbol{D}$, probability $p \in (0,1]$, number of iterations K
Initialize $g^0 = \boldsymbol{D}\cdot\nabla f(x^0)$
for $k = 0, 1, \ldots, K-1$ **do**
  Sample $c_k \sim \text{Be}(p)$
  Broadcast g^k to all workers
  for $i = 1, 2, \ldots$ in parallel **do**
    $x^{k+1} = x^k - g^k$
    Set $g_i^{k+1} = \begin{cases} \boldsymbol{D}\cdot\nabla f_i(x^{k+1}) & \text{if } c_k = 1 \\ g^k + \boldsymbol{T}_i^k\boldsymbol{D}\left(\nabla f_i(x^{k+1}) - \nabla f_i(x^k)\right) & \text{if } c_k = 0 \end{cases}$
  end for
  $g^{k+1} = \frac{1}{n}\sum_{i=1}^n g_i^{k+1}$
end for
Return: $\tilde{x}^K$ chosen uniformly at random from $\{x^k\}_{k=0}^{K-1}$

We call `det-MARINA` as the extension of `det-CGD`, and Algorithm 3 as the extension of `det-CGD2` due to the difference in the order of applying sketches and stepsize matrices. The key difference between `det-CGD` (`det-CGD1`) and `det-CGD2` is that in `det-CGD` the gradient is sketched first and then multiplied by the stepsize, while for `det-CGD2`, the gradient is multiplied by the stepsize first after which the product is sketched. The convergence for Algorithm 3 can be proved in a similar manner as the Theorem 1.

Theorem 3. *Let Assumptions 1 and 3 hold, with the gradient of f being $\boldsymbol{L}$-Lipschitz. If the stepsize matrix $\boldsymbol{D} \in \mathbb{S}^d_{++}$ satisfies*

$$\boldsymbol{D}^{-1} \succeq \left(\frac{(1-p)\cdot R'(\boldsymbol{D},\mathcal{S})}{np} + 1\right)\boldsymbol{L},$$

where

$$R'(\boldsymbol{D},\mathcal{S}) = \frac{1}{n}\sum_{i=1}^n \lambda_{\max}\left(\boldsymbol{D}\mathbb{E}\left[\boldsymbol{T}_i^k\boldsymbol{D}^{-1}\boldsymbol{T}_i^k\right]\boldsymbol{D}\boldsymbol{L}_i^{\frac{1}{2}} - \boldsymbol{L}_i^{\frac{1}{2}}\boldsymbol{D}\right)\cdot\lambda_{\max}\left(\boldsymbol{L}_i\right)\cdot\lambda_{\max}\left(\boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{L}_i\boldsymbol{L}^{-\frac{1}{2}}\right).$$

Then after K iterations of Algorithm 3, we have

$$\mathbb{E}\left[\left\|\nabla f(\tilde{x}^K)\right\|^2_{\frac{\boldsymbol{D}}{\det(\boldsymbol{D})^{1/d}}}\right] \leq \frac{2\left(f(x^0)-f^\star\right)}{\det(\boldsymbol{D})^{1/d}\cdot K}.$$

This is to say that in order to reach a ε-stationary point, we require

$$K \geq \frac{2(f(x^0)-f^\star)}{\det(\boldsymbol{D})^{1/d}\cdot\varepsilon^2}.$$

If we look at the scalar case where $\boldsymbol{D} = \gamma \cdot \boldsymbol{I}_d$, $\boldsymbol{L}_i = L_i \cdot \boldsymbol{I}_d$ and $\boldsymbol{L} = L \cdot I_d$, then the condition in Theorem 3 reduces to

$$\frac{(1-p)\omega L^2}{np} + L - \frac{1}{\gamma} \leq 0. \tag{59}$$

Notice that here $\omega = \lambda_{\max}\left(\mathbb{E}\left[\left(\boldsymbol{T}_i^k\right)^2\right]\right) - 1$, and we have $L^2 = \frac{1}{n}\sum_{i=1}^n L_i^2$, which is due to the relation given in Proposition 4. This condition coincides with the condition for convergence of `MARINA`. One may also check that, the update rule in Algorithm 3, is the same as `MARINA` in the scalar case. However, the condition given in Theorem 3 is not simpler than Theorem 1, contrary to the single-node case. We emphasize that Algorithm 3 is not suitable for the federated learning setting where the clients have limited resources. In order to perform the update, each client is required to store the stepsize matrix $\boldsymbol{D}$ which is of size $d \times d$. In the over-parameterized regime, the dataset size is $m \times d$ where m is the number of data samples, and we have $d > m$. This means that the stepsize matrix each client needs to store is even larger than the dataset itself, which is unacceptable given the limited resources each client has.

G.1 Analysis of Algorithm 3

We first present two lemmas which are necessary for the proofs of Theorem 3.

Lemma 3. *Assume that function f is $\boldsymbol{L}$-smooth, and $x^{k+1} = x^k - g^k$, and matrix $\boldsymbol{D} \in \mathbb{S}^d_{++}$. Then we will have*

$$f(x^{k+1}) \leq f(x^k) - \frac{1}{2}\left\|\nabla f(x^k)\right\|^2_{\boldsymbol{D}} + \frac{1}{2}\left\|\boldsymbol{D}\cdot\nabla f(x^k) - g^k\right\|^2_{\boldsymbol{D}^{-1}} - \frac{1}{2}\left\|x^{k+1}-x^k\right\|^2_{\boldsymbol{D}^{-1}-\boldsymbol{L}}. \tag{60}$$

Note that this lemma is formulated in a different way from Lemma 1 on purpose. It contains the squared norm of difference of g^k and $\boldsymbol{D}\cdot\nabla f(x^k)$, which facilitates the proof of convergence for Algorithm 3.

Lemma 4. *For any sketch matrix $\boldsymbol{T} \in \mathbb{S}^d_+$, vector $t \in \mathbb{R}^d$, matrix $\boldsymbol{D} \in \mathbb{S}^d_{++}$ and matrix $\boldsymbol{L} \in \mathbb{S}^d_{++}$, we have*

$$\mathbb{E}\left[\left\|\boldsymbol{T}\boldsymbol{D}t - \boldsymbol{D}t\right\|^2_{\boldsymbol{D}^{-1}}\right] \leq \lambda_{\max}\left(\boldsymbol{L}^{\frac{1}{2}}\boldsymbol{D}\mathbb{E}\left[\boldsymbol{T}\boldsymbol{D}^{-1}\boldsymbol{T}\right]\boldsymbol{D}\boldsymbol{L}^{\frac{1}{2}} - \boldsymbol{L}^{\frac{1}{2}}\boldsymbol{D}\boldsymbol{L}^{\frac{1}{2}}\right)\|t\|^2_{\boldsymbol{L}^{-1}}. \tag{61}$$

G.2 Proof of Theorem 3

The proof given in this section for the convergence guarantee of Algorithm 3 is similar to the proof of Theorem 1 with some modifications.

Proof of Theorem 3. We start with Lemma 3,

$$\mathbb{E}\left[f(x^{k+1})\right] \leq \mathbb{E}\left[f(x^k)\right] - \mathbb{E}\left[\frac{1}{2}\left\|\nabla f(x^k)\right\|^2_{\boldsymbol{D}}\right] + \mathbb{E}\left[\frac{1}{2}\left\|\boldsymbol{D}\cdot\nabla f(x^k) - g^k\right\|^2_{\boldsymbol{D}^{-1}}\right] - \mathbb{E}\left[\frac{1}{2}\left\|x^{k+1}-x^k\right\|^2_{\boldsymbol{D}^{-1}-\boldsymbol{L}}\right]. \tag{62}$$

Now we do the same as Theorem 1 and look at the term $\mathbb{E}\left[\left\|\boldsymbol{D}\cdot\nabla f(x^{k+1}) - g^{k+1}\right\|^2_{\boldsymbol{D}^{-1}}\right]$. Recall that g^k here is given by

$$g^{k+1} = \begin{cases} \boldsymbol{D}\cdot\nabla f(x^{k+1}) & \text{with probability } p \\ g^k + \frac{1}{n}\sum_{i=1}^n \boldsymbol{T}_i^k \boldsymbol{D}\left(\nabla f_i(x^{k+1}) - \nabla f_i(x^k)\right) & \text{with probability } 1-p\,. \end{cases}$$

As a result, we have

$$
\begin{aligned}
&\mathbb{E}\left[\left\|g^{k+1}-\boldsymbol{D}\nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}^{-1}} \mid x^{k+1},x^k\right]\\
&\quad=\mathbb{E}\left[\mathbb{E}\left[\left\|g^{k+1}-\boldsymbol{D}\nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}^{-1}} \mid x^{k+1},x^k,c_k\right]\right]\\
&\quad=p\cdot\left\|\boldsymbol{D}\nabla f(x^{k+1})-\boldsymbol{D}\nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}^{-1}}\\
&\qquad+(1-p)\cdot\mathbb{E}\left[\left\|g^k+\frac{1}{n}\sum_{i=1}^n \boldsymbol{T}_i^k\boldsymbol{D}\left(\nabla f_i(x^{k+1})-\nabla f_i(x^k)\right)-\boldsymbol{D}\nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}^{-1}} \mid x^{k+1},x^k\right]\\
&\quad=(1-p)\cdot\mathbb{E}\left[\left\|g^k+\frac{1}{n}\sum_{i=1}^n \boldsymbol{T}_i^k\boldsymbol{D}\left(\nabla f_i(x^{k+1})-\nabla f_i(x^k)\right)-\boldsymbol{D}\nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}^{-1}} \mid x^{k+1},x^k\right].
\end{aligned}
$$

Using Fact 2 with $x=\frac{1}{n}\sum_{i=1}^n \boldsymbol{T}_i^k\boldsymbol{D}\left(\nabla f_i(x^{k+1})-\nabla f_i(x^k)\right)$, $c=\boldsymbol{D}\nabla f(x^{k+1})-g^k$, we are able to obtain that

$$
\begin{aligned}
&(1-p)\cdot\mathbb{E}\left[\left\|g^k+\frac{1}{n}\sum_{i=1}^n \boldsymbol{T}_i^k\boldsymbol{D}\left(\nabla f_i(x^{k+1})-\nabla f_i(x^k)\right)-\boldsymbol{D}\nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}^{-1}} \mid x^{k+1},x^k\right]\\
&\quad=(1-p)\cdot\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{T}_i^k\boldsymbol{D}\left(\nabla f_i(x^{k+1})-\nabla f_i(x^k)\right)-\boldsymbol{D}\left(\nabla f(x^{k+1})-\nabla f(x^k)\right)\right\|^2_{\boldsymbol{D}^{-1}} \mid x^{k+1},x^k\right]\\
&\qquad+(1-p)\cdot\left\|g^k-\nabla f(x^k)\right\|^2_{\boldsymbol{D}^{-1}}\\
&\quad=(1-p)\cdot\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \left[\boldsymbol{T}_i^k\boldsymbol{D}\left(\nabla f_i(x^{k+1})-\nabla f_i(x^k)\right)-\boldsymbol{D}\left(\nabla f_i(x^{k+1})-\nabla f_i(x^k)\right)\right]\right\|^2_{\boldsymbol{D}^{-1}} \mid x^{k+1},x^k\right]\\
&\qquad+(1-p)\cdot\left\|g^k-\nabla f(x^k)\right\|^2_{\boldsymbol{D}^{-1}}.
\end{aligned}
$$

It is not hard to notice that for the sketch matrices we pick, the following identity holds due to the unbiasedness,

$$
\mathbb{E}\left[\boldsymbol{T}_i^k\boldsymbol{D}(\nabla f_i(x^{k+1})-\nabla f_i(x^k)) \mid x^{k+1},x^k\right]=\boldsymbol{D}(\nabla f_i(x^{k+1})-\nabla f_i(x^k)),
$$

and any two random vectors in the set $\left\{\boldsymbol{T}_i^k\boldsymbol{D}(\nabla f_i(x^{k+1})-\nabla f_i(x^k))\right\}_{i=1}^n$ are independent if x^{k+1},x^k are fixed. As a result

$$
\begin{aligned}
&\mathbb{E}\left[\left\|g^{k+1}-\boldsymbol{D}\nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}^{-1}} \mid x^{k+1},x^k\right]\\
&\quad=\frac{1-p}{n^2}\sum_{i=1}^n\mathbb{E}\left[\left\|\boldsymbol{T}_i^k\left(\boldsymbol{D}\nabla f_i(x^{k+1})-\boldsymbol{D}\nabla f_i(x^k)\right)-\left(\boldsymbol{D}\nabla f_i(x^{k+1})-\boldsymbol{D}\nabla f_i(x^k)\right)\right\|^2_{\boldsymbol{D}^{-1}} \mid x^{k+1},x^k\right]\\
&\qquad+(1-p)\cdot\left\|g^k-\boldsymbol{D}\nabla f(x^k)\right\|^2_{\boldsymbol{D}^{-1}}. \qquad (63)
\end{aligned}
$$

For each term within the summation, can be further upper bounded using Lemma 4

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\boldsymbol{T}_i^k\left(\boldsymbol{D}\nabla f_i(x^{k+1})-\boldsymbol{D}\nabla f_i(x^k)\right)-\left(\boldsymbol{D}\nabla f_i(x^{k+1})-\boldsymbol{D}\nabla f_i(x^k)\right)\right\|^2_{\boldsymbol{D}^{-1}} \mid x^{k+1},x^k\right]\\
&\quad\le\lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}}\boldsymbol{D}\mathbb{E}\left[\boldsymbol{T}_i^k\boldsymbol{D}^{-1}\boldsymbol{T}_i^k\right]\boldsymbol{D}\boldsymbol{L}_i^{\frac{1}{2}}-\boldsymbol{L}_i^{\frac{1}{2}}\boldsymbol{D}\boldsymbol{L}_i^{\frac{1}{2}}\right)\left\|\nabla f_i(x^{k+1})-\nabla f_i(x^k)\right\|^2_{\boldsymbol{L}_i^{-1}}\\
&\quad\le\lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}}\boldsymbol{D}\mathbb{E}\left[\boldsymbol{T}_i^k\boldsymbol{D}^{-1}\boldsymbol{T}_i^k\right]\boldsymbol{D}\boldsymbol{L}_i^{\frac{1}{2}}-\boldsymbol{L}_i^{\frac{1}{2}}\boldsymbol{D}\boldsymbol{L}_i^{\frac{1}{2}}\right)\left\|x^{k+1}-x^k\right\|^2_{\boldsymbol{L}_i}.
\end{aligned}
$$

Where the last inequality is due to Assumption 3. Plugging back into (63), we get

$$
\begin{aligned}
&\mathbb{E}\left[\left\|g^{k+1}-\boldsymbol{D}\nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}^{-1}} \mid x^{k+1},x^k\right]\\
&\quad\le\frac{1-p}{n^2}\sum_{i=1}^n\lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}}\boldsymbol{D}\mathbb{E}\left[\boldsymbol{T}_i^k\boldsymbol{D}^{-1}\boldsymbol{T}_i^k\right]\boldsymbol{D}\boldsymbol{L}_i^{\frac{1}{2}}-\boldsymbol{L}_i^{\frac{1}{2}}\boldsymbol{D}\boldsymbol{L}_i^{\frac{1}{2}}\right)\left\|x^{k+1}-x^k\right\|^2_{\boldsymbol{L}_i}\\
&\qquad+(1-p)\cdot\left\|g^k-\boldsymbol{D}\nabla f(x^k)\right\|^2_{\boldsymbol{D}^{-1}}
\end{aligned}
$$

Applying the replacement trick form the proof of Theorem 1, we obtain

$$\begin{aligned}
&\mathbb{E}\left[\left\|g^{k+1}-\boldsymbol{D}\nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}^{-1}} \mid x^{k+1},x^k\right] \\
&\quad\leq \frac{1-p}{n^2}\sum_{i=1}^n \lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}}\boldsymbol{D}\mathbb{E}\left[\boldsymbol{T}_i^k\boldsymbol{D}^{-1}\boldsymbol{T}_i^k\right]\boldsymbol{D}\boldsymbol{L}_i^{\frac{1}{2}}-\boldsymbol{L}_i^{\frac{1}{2}}\boldsymbol{D}\boldsymbol{L}_i^{\frac{1}{2}}\right) \\
&\qquad\times\left\langle \boldsymbol{L}^{\frac{1}{2}}\left(x^{k+1}-x^k\right),\left(\boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{L}_i\boldsymbol{L}^{-\frac{1}{2}}\right)\cdot\boldsymbol{L}^{\frac{1}{2}}\left(x^{k+1}-x^k\right)\right\rangle+(1-p)\cdot\left\|g^k-\boldsymbol{D}\nabla f(x^k)\right\|^2_{\boldsymbol{D}^{-1}} \\
&\quad\leq \frac{1-p}{n^2}\sum_{i=1}^n \lambda_{\max}\left(\boldsymbol{L}_i^{\frac{1}{2}}\boldsymbol{D}\mathbb{E}\left[\boldsymbol{T}_i^k\boldsymbol{D}^{-1}\boldsymbol{T}_i^k\right]\boldsymbol{D}\boldsymbol{L}_i^{\frac{1}{2}}-\boldsymbol{L}_i^{\frac{1}{2}}\boldsymbol{D}\boldsymbol{L}_i^{\frac{1}{2}}\right)\cdot\lambda_{\max}\left(\boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{L}_i\boldsymbol{L}^{-\frac{1}{2}}\right)\left\|x^{k+1}-x^k\right\|^2_{\boldsymbol{L}} \\
&\qquad+(1-p)\cdot\left\|g^k-\boldsymbol{D}\nabla f(x^k)\right\|^2_{\boldsymbol{D}^{-1}}.
\end{aligned}$$

Applying Fact 5, we obtain

$$\begin{aligned}
&\mathbb{E}\left[\left\|g^{k+1}-\boldsymbol{D}\nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}^{-1}} \mid x^{k+1},x^k\right] \\
&\quad\leq \frac{1-p}{n^2}\sum_{i=1}^n \lambda_{\max}\left(\boldsymbol{D}\mathbb{E}\left[\boldsymbol{T}_i^k\boldsymbol{D}^{-1}\boldsymbol{T}_i^k\right]\boldsymbol{D}-\boldsymbol{D}\right)\cdot\lambda_{\max}\left(\boldsymbol{L}_i\right)\cdot\lambda_{\max}\left(\boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{L}_i\boldsymbol{L}^{-\frac{1}{2}}\right)\left\|x^{k+1}-x^k\right\|^2_{\boldsymbol{L}} \\
&\qquad+(1-p)\cdot\left\|g^k-\boldsymbol{D}\nabla f(x^k)\right\|^2_{\boldsymbol{D}^{-1}}.
\end{aligned}$$

Recalling the definition of $R'(\boldsymbol{D},\mathcal{S})$, the previous upper bound further simplifies to

$$\begin{aligned}
&\mathbb{E}\left[\left\|g^{k+1}-\boldsymbol{D}\nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}^{-1}} \mid x^{k+1},x^k\right] \\
&\quad\leq \frac{(1-p)\cdot R'(\boldsymbol{D},\mathcal{S})}{n}\left\|x^{k+1}-x^k\right\|^2_{\boldsymbol{L}}+(1-p)\cdot\left\|g^k-\boldsymbol{D}\nabla f(x^k)\right\|^2_{\boldsymbol{D}^{-1}}.
\end{aligned}$$

Taking expectation again and using the tower property, we get

$$\begin{aligned}
&\mathbb{E}\left[\left\|g^{k+1}-\boldsymbol{D}\nabla f(x^{k+1})\right\|^2_{\boldsymbol{D}^{-1}}\right] \\
&\quad\leq \frac{(1-p)\cdot R'(\boldsymbol{D},\mathcal{S})}{n}\mathbb{E}\left[\left\|x^{k+1}-x^k\right\|^2_{\boldsymbol{L}}\right]+(1-p)\cdot\mathbb{E}\left[\left\|g^k-\boldsymbol{D}\nabla f(x^k)\right\|^2_{\boldsymbol{D}^{-1}}\right]. \qquad (64)
\end{aligned}$$

Construct the Lyapunov function Φ_k as follows,

$$\Phi_k=f(x^k)-f^\star+\frac{1}{2p}\left\|g^k-\boldsymbol{D}\nabla f(x^k)\right\|^2_{\boldsymbol{D}^{-1}}.$$

Utilizing (62) and (64), we are able to get

$$\begin{aligned}
\mathbb{E}\left[\Phi_{k+1}\right] &\leq \mathbb{E}\left[f(x^k)-f^\star\right]-\frac{1}{2}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2_{\boldsymbol{D}}\right]+\frac{1}{2}\mathbb{E}\left[\left\|g^k-\boldsymbol{D}\nabla f(x^k)\right\|^2_{\boldsymbol{D}^{-1}}\right]-\frac{1}{2}\mathbb{E}\left[\left\|x^{k+1}-x^k\right\|_{\boldsymbol{D}^{-1}-L}\right] \\
&\quad+\frac{1}{2p}\cdot\frac{(1-p)R'(\boldsymbol{D},\mathcal{S})}{n}\mathbb{E}\left[\left\|x^{k+1}-x^k\right\|^2_{\boldsymbol{L}}\right]+\frac{1-p}{2p}\mathbb{E}\left[\left\|g^k-\boldsymbol{D}\nabla f(x^k)\right\|^2_{\boldsymbol{D}^{-1}}\right] \\
&=\mathbb{E}\left[\Phi_k\right]-\frac{1}{2}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2_{\boldsymbol{D}}\right] \\
&\quad+\frac{1}{2}\left(\frac{(1-p)R'(\boldsymbol{D},\mathcal{S})}{np}\mathbb{E}\left[\left\|x^{k+1}-x^k\right\|^2_{\boldsymbol{L}}\right]-\mathbb{E}\left[\left\|x^{k+1}-x^k\right\|^2_{\boldsymbol{D}^{-1}-\boldsymbol{L}}\right]\right).
\end{aligned}$$

Now, notice that the last term in the above inequality is non-positive as guaranteed by the condition

$$\boldsymbol{D}^{-1}\succeq\left(\frac{(1-p)R'(\boldsymbol{D},\mathcal{S})}{np}+1\right)\boldsymbol{L}.$$

This leads to the recurrence after ignoring the last term,

$$\mathbb{E}\left[\Phi_{k+1}\right]\leq\mathbb{E}\left[\Phi_k\right]-\frac{1}{2}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2_{\boldsymbol{D}}\right].$$

Unrolling this recurrence relation, we get

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|_{\boldsymbol{D}}^2\right] \leq \frac{2\left(\mathbb{E}\left[\Phi_0\right]-\mathbb{E}\left[\Phi_K\right]\right)}{K}. \tag{65}$$

The left hand side can viewed as average over $\tilde{x}^K$, which is drawn uniformly at random from $\{x_k\}_{k=0}^{K-1}$, while the right hand side can be simplified as

$$\begin{aligned}\frac{2\left(\mathbb{E}\left[\Phi_0\right]-\mathbb{E}\left[\Phi_K\right]\right)}{K} &\leq \frac{2\Phi_0}{K}\\ &= \frac{2\left(f(x^0)-f^\star+\frac{1}{2p}\left\|g^0-\nabla f(x^0)\right\|_{\boldsymbol{D}}^2\right)}{K}.\end{aligned}$$

Recalling that $g^0=\nabla f(x^0)$ and performing determinant normalization as [LKR23], we get

$$\mathbb{E}\left[\left\|\nabla f(\tilde{x}^K)\right\|^2_{\frac{\boldsymbol{D}}{\det(\boldsymbol{D})^{1/d}}}\right] \leq \frac{2\left(f(x^0)-f^\star\right)}{\det(\boldsymbol{D})^{1/d}K}.$$

This completes the proof. □

H Proofs of the technical lemmas

H.1 Proof of Lemma 1

Let $\bar{x}^{k+1}:=x^k-\boldsymbol{D}\cdot\nabla f(x^k)$. Since f has matrix $\boldsymbol{L}$-Lipschitz gradient, f is also $\boldsymbol{L}$-smooth. From the $\boldsymbol{L}$-smoothness of f, we have

$$\begin{aligned}f(x^{k+1}) &\leq f(x^k)+\left\langle\nabla f(x^k),x^{k+1}-x^k\right\rangle+\frac{1}{2}\left\langle x^{k+1}-x^k,\boldsymbol{L}(x^{k+1}-x^k)\right\rangle\\ &= f(x^k)+\left\langle\nabla f(x^k)-g^k,x^{k+1}-x^k\right\rangle+\left\langle g^k,x^{k+1}-x^k\right\rangle+\frac{1}{2}\left\langle x^{k+1}-x^k,\boldsymbol{L}(x^{k+1}-x^k)\right\rangle.\end{aligned}$$

We can merge the last two terms and obtain,

$$\begin{aligned}f(x^{k+1}) &\leq f(x^k)+\left\langle\nabla f(x^k)-g^k,-\boldsymbol{D}\cdot g^k\right\rangle-\left\langle x^{k+1}-x^k,\boldsymbol{D}^{-1}(x^{k+1}-x^k)\right\rangle\\ &\quad+\frac{1}{2}\left\langle x^{k+1}-x^k,\boldsymbol{L}(x^{k+1}-x^k)\right\rangle\\ &= f(x^k)+\left\langle\nabla f(x^k)-g^k,-\boldsymbol{D}\cdot g^k\right\rangle-\left\langle x^{k+1}-x^k,\left(\boldsymbol{D}^{-1}-\frac{1}{2}\boldsymbol{L}\right)(x^{k+1}-x^k)\right\rangle.\end{aligned}$$

We add and subtract $\left\langle\nabla f(x^k)-g^k,\boldsymbol{D}\cdot g^k\right\rangle$,

$$\begin{aligned}f(x^{k+1}) &\leq f(x^k)+\left\langle\nabla f(x^k)-g^k,\boldsymbol{D}\left(\nabla f(x^k)-g^k\right)\right\rangle-\left\langle\nabla f(x^k)-g^k,\boldsymbol{D}\cdot\nabla f(x^k)\right\rangle\\ &\quad-\left\langle x^{k+1}-x^k,\left(\boldsymbol{D}^{-1}-\frac{1}{2}\boldsymbol{L}\right)(x^{k+1}-x^k)\right\rangle\\ &= f(x^k)+\left\|\nabla f(x^k)-g^k\right\|_{\boldsymbol{D}}^2-\left\langle x^{k+1}-\bar{x}^{k+1},\boldsymbol{D}^{-1}\left(x^k-\bar{x}^{k+1}\right)\right\rangle\\ &\quad-\left\langle x^{k+1}-x^k,\left(\boldsymbol{D}^{-1}-\frac{1}{2}\boldsymbol{L}\right)(x^{k+1}-x^k)\right\rangle.\end{aligned}$$

Decomposing the term $\left\langle x^{k+1}-\bar{x}^{k+1},\boldsymbol{D}^{-1}\left(x^k-\bar{x}^{k+1}\right)\right\rangle$, we get

$$\begin{aligned}f(x^{k+1}) &\leq f(x^k)+\left\|\nabla f(x^k)-g^k\right\|_{\boldsymbol{D}}^2-\left\langle x^{k+1}-x^k,\left(\boldsymbol{D}^{-1}-\frac{1}{2}\boldsymbol{L}\right)(x^{k+1}-x^k)\right\rangle\\ &\quad-\frac{1}{2}\left(\left\|x^{k+1}-\bar{x}^{k+1}\right\|_{\boldsymbol{D}^{-1}}^2+\left\|x^k-\bar{x}^{k+1}\right\|_{\boldsymbol{D}^{-1}}^2-\left\|x^{k+1}-x^k\right\|_{\boldsymbol{D}^{-1}}^2\right).\end{aligned}$$

Plugging in the definition of $x^{k+1}, \bar{x}^{k+1}$, we get

$$\begin{aligned} f(x^{k+1}) &\leq f(x^k) + \left\|\nabla f(x^k) - g^k\right\|_{\boldsymbol{D}}^2 - \left\|x^{k+1} - x^k\right\|_{\boldsymbol{D}^{-1} - \frac{1}{2}\boldsymbol{L}}^2 \\ &\quad -\frac{1}{2}\left(\left\|\boldsymbol{D}(\nabla f(x^k) - g^k)\right\|_{\boldsymbol{D}^{-1}}^2 + \left\|\boldsymbol{D}\cdot\nabla f(x^k)\right\|_{\boldsymbol{D}^{-1}}^2 - \left\|x^{k+1} - x^k\right\|_{\boldsymbol{D}^{-1}}^2\right) \\ &= f(x^k) + \left\|\nabla f(x^k) - g^k\right\|_{\boldsymbol{D}}^2 - \left\|x^{k+1} - x^k\right\|_{\boldsymbol{D}^{-1} - \frac{1}{2}\boldsymbol{L}}^2 \\ &\quad -\frac{1}{2}\left(\left\|\nabla f(x^k) - g^k\right\|_{\boldsymbol{D}}^2 + \left\|\nabla f(x^k)\right\|_{\boldsymbol{D}}^2 - \left\|x^{k+1} - x^k\right\|_{\boldsymbol{D}^{-1}}^2\right). \end{aligned}$$

Rearranging terms we get,

$$\begin{aligned} f(x^{k+1}) &\leq f(x^k) - \frac{1}{2}\left\|\nabla f(x^k)\right\|_{\boldsymbol{D}}^2 + \frac{1}{2}\left\|g^k - \nabla f(x^k)\right\|_{\boldsymbol{D}}^2 - \left\|x^{k+1} - x^k\right\|_{\boldsymbol{D}^{-1} - \frac{1}{2}\boldsymbol{L}}^2 + \frac{1}{2}\left\|x^{k+1} - x^k\right\|_{\boldsymbol{D}^{-1}}^2 \\ &= f(x^k) - \frac{1}{2}\left\|\nabla f(x^k)\right\|_{\boldsymbol{D}}^2 + \frac{1}{2}\left\|g^k - \nabla f(x^k)\right\|_{\boldsymbol{D}}^2 - \frac{1}{2}\left\|x^{k+1} - x^k\right\|_{\boldsymbol{D}^{-1} - \boldsymbol{L}}. \end{aligned}$$

This completes the proof.

H.2 Proof of Lemma 2

The definition of the weighted norm yields

$$\begin{aligned} \mathbb{E}\left[\left\|\boldsymbol{S}t - t\right\|_{\boldsymbol{D}}^2\right] &= \mathbb{E}\left[\langle t, (\boldsymbol{S} - \boldsymbol{I}_d)\,\boldsymbol{D}\,(\boldsymbol{S} - \boldsymbol{I}_d)\,t\rangle\right] \\ &= \langle t, \mathbb{E}\left[(\boldsymbol{S} - \boldsymbol{I}_d)\boldsymbol{D}(\boldsymbol{S} - \boldsymbol{I}_d)\right] t\rangle \\ &= \left\langle t, \boldsymbol{L}^{-\frac{1}{2}}\cdot\mathbb{E}\left[\boldsymbol{L}^{\frac{1}{2}}(\boldsymbol{S} - \boldsymbol{I}_d)\boldsymbol{D}(\boldsymbol{S} - \boldsymbol{I}_d)\boldsymbol{L}^{\frac{1}{2}}\right]\cdot\boldsymbol{L}^{-\frac{1}{2}}t\right\rangle \\ &= \left\langle \boldsymbol{L}^{-\frac{1}{2}}t, \mathbb{E}\left[\boldsymbol{L}^{\frac{1}{2}}(\boldsymbol{S} - \boldsymbol{I}_d)\boldsymbol{D}(\boldsymbol{S} - \boldsymbol{I}_d)\boldsymbol{L}^{\frac{1}{2}}\right]\cdot\boldsymbol{L}^{-\frac{1}{2}}t\right\rangle \\ &\leq \lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{L}^{\frac{1}{2}}(\boldsymbol{S} - \boldsymbol{I}_d)\boldsymbol{D}(\boldsymbol{S} - \boldsymbol{I}_d)\boldsymbol{L}^{\frac{1}{2}}\right]\right)\left\|\boldsymbol{L}^{-\frac{1}{2}}t\right\|^2 \\ &= \lambda_{\max}\left(\boldsymbol{L}^{\frac{1}{2}}\left(\mathbb{E}\left[\boldsymbol{S}\boldsymbol{D}\boldsymbol{S}\right] - \boldsymbol{D}\right)\boldsymbol{L}^{\frac{1}{2}}\right)\cdot\left\|t\right\|_{\boldsymbol{L}^{-1}}^2 \end{aligned}$$

This completes the proof.

H.3 Proof of Lemma 3

Let $\bar{x}^{k+1} = x^k - \boldsymbol{D}\cdot\nabla f(x^k)$, we start with the definition of $\boldsymbol{L}$-smoothness. Notice that $\boldsymbol{L}$-smoothness of f is a direct result of matrix $\boldsymbol{L}$-Lipschitz gradient of f.

$$\begin{aligned} f(x^{k+1}) &\leq f(x^k) + \left\langle\nabla f(x^k), x^{k+1} - x^k\right\rangle + \frac{1}{2}\left\langle x^{k+1} - x^k, \boldsymbol{L}(x^{k+1} - x^k)\right\rangle \\ &= f(x^k) + \left\langle\nabla f(x^k) - \boldsymbol{D}^{-1}\cdot g^k, x^{k+1} - x^k\right\rangle + \left\langle\boldsymbol{D}^{-1}\cdot g^k, x^{k+1} - x^k\right\rangle \\ &\quad + \frac{1}{2}\left\langle x^{k+1} - x^k, \boldsymbol{L}(x^{k+1} - x^k)\right\rangle \\ &= f(x^k) + \left\langle\nabla f(x^k) - \boldsymbol{D}^{-1}\cdot g^k, -g^k\right\rangle - \left\langle x^{k+1} - x^k, \boldsymbol{D}^{-1}(x^{k+1} - x^k)\right\rangle \\ &\quad + \frac{1}{2}\left\langle x^{k+1} - x^k, \boldsymbol{L}(x^{k+1} - x^k)\right\rangle. \end{aligned}$$

Simplifying the above inner-products we have,

$$f(x^{k+1}) \leq f(x^k) + \left\langle\nabla f(x^k) - \boldsymbol{D}^{-1}\cdot g^k, -g^k\right\rangle - \left\langle x^{k+1} - x^k, \left(\boldsymbol{D}^{-1} - \frac{1}{2}\boldsymbol{L}\right)(x^{k+1} - x^k)\right\rangle.$$

We then add and subtract $\left\langle \nabla f(x^k) - \boldsymbol{D}^{-1} \cdot g^k, \boldsymbol{D} \cdot \nabla f(x^k) \right\rangle$, and get

$$\begin{aligned} f(x^{k+1}) &\leq f(x^k) + \left\langle \nabla f(x^k) - \boldsymbol{D}^{-1} \cdot g^k, \boldsymbol{D} \cdot \nabla f(x^k) - g^k \right\rangle - \left\langle \nabla f(x^k) - \boldsymbol{D}^{-1} \cdot g^k, \boldsymbol{D} \cdot \nabla f(x^k) \right\rangle \\ &\quad - \left\langle x^{k+1} - x^k, \left(\boldsymbol{D}^{-1} - \frac{1}{2}\boldsymbol{L} \right) (x^{k+1} - x^k) \right\rangle \\ &= f(x^k) + \left\| \nabla f(x^k) - \boldsymbol{D}^{-1} \cdot g^k \right\|_{\boldsymbol{D}}^2 - \left\langle \boldsymbol{D}^{-1}(x^{k+1} - \bar{x}^{k+1}), x^k - \bar{x}^{k+1} \right\rangle \\ &\quad - \left\langle x^{k+1} - x^k, \left(\boldsymbol{D}^{-1} - \frac{1}{2}\boldsymbol{L} \right) (x^{k+1} - x^k) \right\rangle. \end{aligned}$$

Decomposing the inner product term we get,

$$\begin{aligned} f(x^{k+1}) &\leq f(x^k) + \left\| \boldsymbol{D}^{-1} \left(\boldsymbol{D} \cdot \nabla f(x^k) - g^k \right) \right\|_{\boldsymbol{D}}^2 - \left\langle x^{k+1} - x^k, \left(\boldsymbol{D}^{-1} - \frac{1}{2}\boldsymbol{L} \right) (x^{k+1} - x^k) \right\rangle \\ &\quad - \frac{1}{2} \left(\left\| x^{k+1} - \bar{x}^{k+1} \right\|_{\boldsymbol{D}^{-1}}^2 + \left\| x^k - \bar{x}^{k+1} \right\|_{\boldsymbol{D}^{-1}}^2 - \left\| x^{k+1} - x^k \right\|_{\boldsymbol{D}^{-1}}^2 \right) \\ &= f(x^k) + \left\| \boldsymbol{D} \cdot \nabla f(x^k) - g^k \right\|_{\boldsymbol{D}^{-1}}^2 - \left\| x^{k+1} - x^k \right\|_{\boldsymbol{D}^{-1} - \frac{1}{2}\boldsymbol{L}}^2 \\ &\quad - \frac{1}{2} \left(\left\| \boldsymbol{D} \cdot \nabla f(x^k) - g^k \right\|_{\boldsymbol{D}^{-1}}^2 + \left\| \boldsymbol{D} \cdot \nabla f(x^k) \right\|_{\boldsymbol{D}^{-1}}^2 - \left\| x^{k+1} - x^k \right\|_{\boldsymbol{D}^{-1}}^2 \right). \end{aligned}$$

Rearranging terms we get,

$$f(x^{k+1}) \leq f(x^k) + \frac{1}{2} \left\| \boldsymbol{D} \nabla f(x^k) - g^k \right\|_{\boldsymbol{D}^{-1}}^2 - \frac{1}{2} \left\| \nabla f(x^k) \right\|_{\boldsymbol{D}}^2 - \frac{1}{2} \left\| x^{k+1} - x^k \right\|_{\boldsymbol{D}^{-1} - \boldsymbol{L}}^2.$$

This completes the proof.

H.4 Proof of Lemma 4

We start with

$$\begin{aligned} \mathbb{E}\left[\left\| \boldsymbol{T}\boldsymbol{D}t - \boldsymbol{D}t \right\|_{\boldsymbol{D}^{-1}}^2 \right] &= \mathbb{E}\left[\left\| (\boldsymbol{T} - \boldsymbol{I}_d)\boldsymbol{D}t \right\|_{\boldsymbol{D}^{-1}}^2 \right] \\ &= \left\langle t, \mathbb{E}\left[\boldsymbol{D}(\boldsymbol{T} - \boldsymbol{I}_d)\boldsymbol{D}^{-1}(\boldsymbol{T} - \boldsymbol{I}_d)\boldsymbol{D} \right] \cdot t \right\rangle \\ &= \left\langle t, \boldsymbol{D}\left(\mathbb{E}\left[\boldsymbol{T}\boldsymbol{D}^{-1}\boldsymbol{T} \right] - \boldsymbol{D}^{-1} \right) \boldsymbol{D} \cdot t \right\rangle \\ &= \left\langle \boldsymbol{L}^{-\frac{1}{2}} t, \boldsymbol{L}^{\frac{1}{2}} \boldsymbol{D}\left(\mathbb{E}\left[\boldsymbol{T}\boldsymbol{D}^{-1}\boldsymbol{T} \right] - \boldsymbol{D}^{-1} \right) \boldsymbol{D}\boldsymbol{L}^{\frac{1}{2}} \cdot \boldsymbol{L}^{-\frac{1}{2}} t \right\rangle \\ &\leq \lambda_{\max}\left(\boldsymbol{L}^{\frac{1}{2}} \boldsymbol{D}\mathbb{E}\left[\boldsymbol{T}\boldsymbol{D}^{-1}\boldsymbol{T} \right] \boldsymbol{D}\boldsymbol{L}^{\frac{1}{2}} - \boldsymbol{L}^{\frac{1}{2}} \boldsymbol{D}\boldsymbol{L}^{\frac{1}{2}} \right) \cdot \left\| \boldsymbol{L}^{-\frac{1}{2}} t \right\|^2 \\ &= \lambda_{\max}\left(\boldsymbol{L}^{\frac{1}{2}} \boldsymbol{D}\mathbb{E}\left[\boldsymbol{T}\boldsymbol{D}^{-1}\boldsymbol{T} \right] \boldsymbol{D}\boldsymbol{L}^{\frac{1}{2}} - \boldsymbol{L}^{\frac{1}{2}} \boldsymbol{D}\boldsymbol{L}^{\frac{1}{2}} \right) \cdot \|t\|_{\boldsymbol{L}^{-1}}^2 \end{aligned}$$

This completes the proof.

I Experiments

In this section, we conduct numerical experiments on the performance of `det-MARINA`. The code for the experiments can be found in https://anonymous.4open.science/r/detMARINA-7D05/.

I.1 The Setting

We first state the experiment setting. We are interested in the following logistic regression problem with a non-convex regularizer. The objective is given as

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x); \qquad f_i(x) = \frac{1}{m_i} \sum_{j=1}^{m_i} \log\left(1 + e^{-b_{i,j} \cdot \langle a_{i,j}, x \rangle} \right) + \lambda \cdot \sum_{t=1}^{d} \frac{x_t^2}{1 + x_t^2},$$

where $x \in \mathbb{R}^d$ is the model, $(a_{i,j}, b_{i,j}) \in \mathbb{R}^d \times \{-1, 1\}$ is one data point in the dataset of client i whose size is m_i. The constant $\lambda > 0$ is the coefficient of the regularizer. Larger λ means the model is more regular. For each function f_i, its Hessian can be upper bounded by

$$\boldsymbol{L}_i = \frac{1}{m_i} \sum_{i=1}^{m_i} \frac{a_i a_i^\top}{4} + 2\lambda \cdot \boldsymbol{I}_d;$$

and, therefore, the Hessian of f is bounded by

$$\boldsymbol{L} = \frac{1}{\sum_{i=1}^n m_i} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \frac{a_i a_i^\top}{4} + 2\lambda \cdot \boldsymbol{I}_d.$$

Due to Proposition 1, it immediately follows that f_i and f satisfy Definition 1 with $\boldsymbol{L}_i \in \mathbb{S}^d_{++}$ and $\boldsymbol{L} \in \mathbb{S}^d_{++}$, respectively. In the following subsections, we perform several numerical experiments comparing the performance of `DCGD`, `det-CGD`, `MARINA` and `det-MARINA`. The datasets we used are from the LibSVM repository ([CL11].)

I.2 Improvements on `MARINA`

The purpose of this experiment is to compare the iteration complexity of `MARINA` with `det-MARINA` using Rand-τ sketches, thus showing improvements of `det-MARINA` upon `MARINA`. Using the Theorem C.1 from [GBLR21], we can deduce the optimal stepsize for `MARINA` is

$$\gamma_1 = \frac{1}{L\left(1 + \sqrt{\frac{(1-p)\omega}{pn}}\right)}, \tag{66}$$

where ω is the quantization coefficient. In particular, for the Rand-τ compressor is $\frac{d}{\tau} - 1$. For the full definition see [GBLR21, Section 1.3]. The stepsize for `det-MARINA` is determined through Corollary 1. We use the notation $\boldsymbol{D}^*_{\boldsymbol{W}}$ to denote the optimal stepsize for each choice of $\boldsymbol{W}$, here we list some of the optimal stepsizes for different $\boldsymbol{W}$, which are used in the experiment section. We have

$$\begin{aligned}
\boldsymbol{D}^*_{\boldsymbol{I}_d} &= \frac{2}{1 + \sqrt{1 + 4\alpha\beta \frac{1}{\lambda_{\max}(\boldsymbol{L})} \cdot \omega}} \cdot \frac{\boldsymbol{I}_d}{\lambda_{\max}(\boldsymbol{L})}, \\
\boldsymbol{D}^*_{\boldsymbol{L}^{-1}} &= \frac{2}{1 + \sqrt{1 + 4\alpha\beta \cdot \lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k \boldsymbol{L}^{-1} \boldsymbol{S}_i^k\right] - \boldsymbol{L}^{-1}\right)}} \cdot \boldsymbol{L}^{-1}, \\
\boldsymbol{D}^*_{\operatorname{diag}^{-1}(\boldsymbol{L})} &= \frac{2}{1 + \sqrt{1 + 4\alpha\beta \cdot \lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{S}_i^k \operatorname{diag}^{-1}(\boldsymbol{L}) \boldsymbol{S}_i^k\right] - \operatorname{diag}^{-1}(\boldsymbol{L})\right)}} \cdot \operatorname{diag}^{-1}(\boldsymbol{L})
\end{aligned} \tag{67}$$

In this experiment, we aim to compare `det-MARINA` with stepsize $\boldsymbol{D}^*_{\boldsymbol{L}^{-1}}$ to standard `MARINA` with optimal scalar stepsize. Rand-τ compressor is used in the comparison. Throughout the experiments, λ is fixed at 0.3. We set the x-axis to be the number of iterations, while y-axis to be the expectation of the corresponding matrix norm of the gradient of the function, which is defined as

$$G_{K,\boldsymbol{D}} = \mathbb{E}\left[\left\|\nabla f(\tilde{x}^K)\right\|^2_{\boldsymbol{D}/\det(\boldsymbol{D})^{1/d}}\right]. \tag{68}$$

As it is illustrated in Figure 1, `det-MARINA` always has a faster convergence rate compared to `MARINA` if they use the same sketch, this justifies the result we have in Corollary 2. Notice that in some cases, `det-MARINA` with rand-1 sketch even outperforms standard `MARINA` with rand-80 sketch. This further demonstrates the superiority of matrix stepsizes and smoothness over the standard scalar setting.

I.3 Improvements on non Variance Reduced Methods

In this section, we compare two non-variance reduced methods, distributed compressed gradient descent (`DCGD`) and distributed `det-CGD` with two variance reduced methods, `MARINA` and `det-MARINA`. Rand-1 sketch is used throughout this experiment for all the algorithms, for non variance reduced

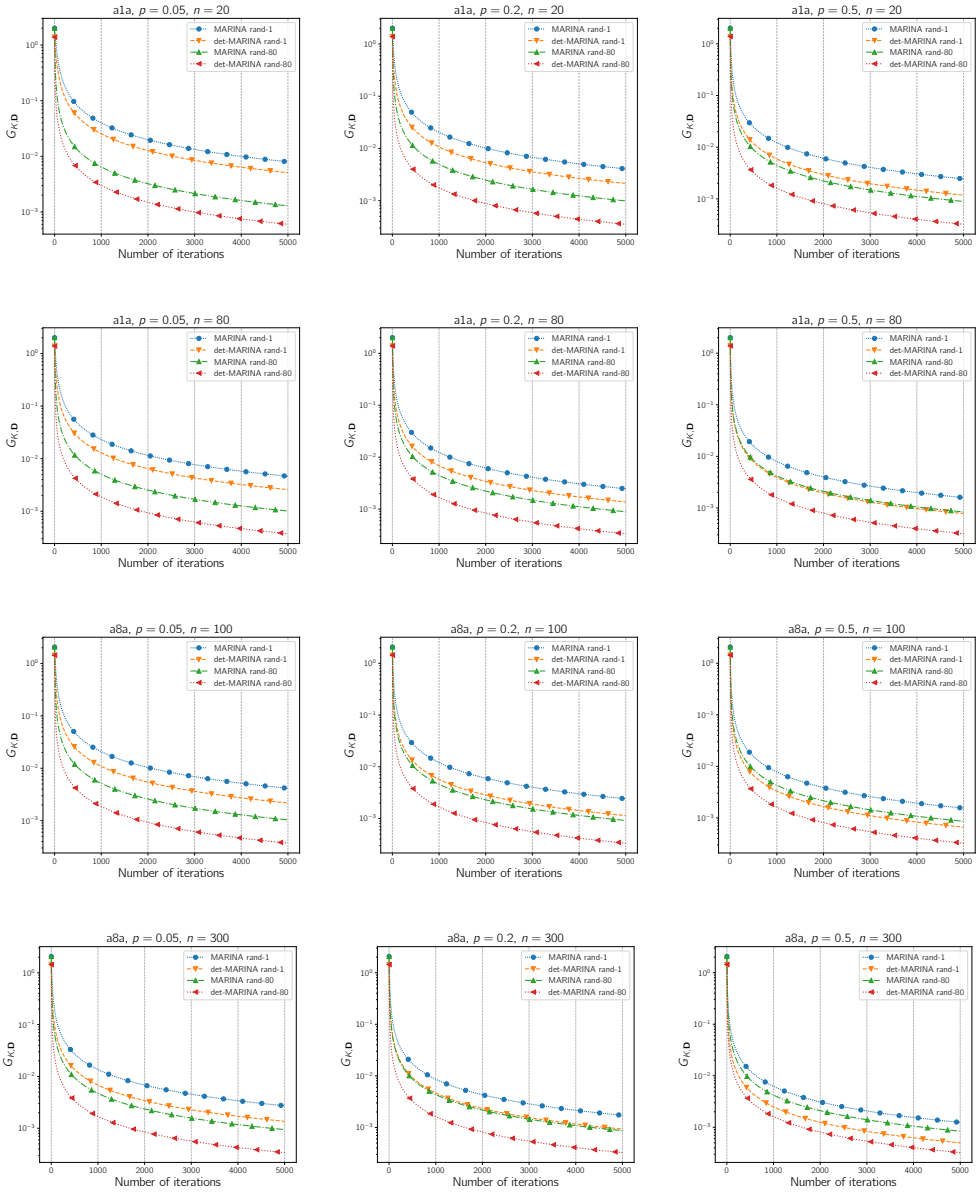

Figure 1: Comparison of `det-MARINA` with matrix stepsize $\boldsymbol{D}^*_{\boldsymbol{L}^{-1}}$ and `MARINA` with optimal scalar stepsize γ_1 using different rand-τ sketches. $\lambda = 0.3$ is fixed throughout the experiments. Optimal stepsize is calculated in each case with respect to the sketch used. The x-axis denotes the number of iterations while the notation $G_{K,\boldsymbol{D}}$ for the y-axis is defined in (68), which is the averaged matrix norm of the gradient. The notation p in the title denotes the probability used in the two algorithms, n denotes the number of clients in each setting.

method ε^2 is fixed at 0.01 in order to determine the optimal stepsize. The purpose of this experiment is to show the advantages of variance reduced methods over non variance reduced methods. DCGD was initially proposed in [KFJ18]. Later on DIANA was proposed in [MGTR19] and then combined with variance reduction technique. Recently [SR22] proposed DCGD-SHIFT, which is a shifted version of DCGD and proved its convergence in the (strongly) convex setting. A general analysis on SGD type methods in the non-convex world is provided by [KR20], including DCGD and DCGD-SHIFT. In our case, in order to determine the optimal scalar stepsize for DCGD, one can simply use Proposition 4 in [KR20]. One can check that in order to satisfy $\min_{0\leq k\leq K-1}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2\right]\leq\varepsilon^2$ the stepsize condition for DCGD in the non-convex case reduces to

$$\gamma_2\leq\min\left\{\frac{1}{L},\sqrt{\frac{n}{\omega LL_{\max}K}},\frac{n\varepsilon^2}{4LL_{\max}\omega\cdot\Delta^\star}\right\},\tag{69}$$

where L is the smoothness constant for f, L_i is the smoothness constant for f_i, $L_{\max}=\max_i L_i$, K is the total number of iterations, $\Delta^\star=f(x^\star)-\frac{1}{n}\sum_{i=1}^n f_i(x^\star)$. The constant ω is associated with the compressor used in the algorithm, for Rand-τ sketch, it is $\frac{d}{\tau}-1$. For distributed det-CGD according to [LKR23], the stepsize condition in order to satisfy $\min_{0\leq k\leq K-1}\mathbb{E}\left[\|\nabla f(x)\|^2_{\boldsymbol{D}/\det(\boldsymbol{D})^{1/d}}\right]\leq\varepsilon^2$ is

$$\boldsymbol{DLD}\preceq\boldsymbol{D},\qquad\lambda_{\boldsymbol{D}}\leq\min\left\{\frac{n}{K},\frac{n\varepsilon^2}{4\Delta^\star}\det(\boldsymbol{D})^{1/d}\right\},\tag{70}$$

where $\lambda_{\boldsymbol{D}}$ is defined as

$$\lambda_{\boldsymbol{D}}=\max_i\left\{\lambda_{\max}\left(\mathbb{E}\left[\boldsymbol{L}_i^{\frac{1}{2}}\left(\boldsymbol{S}_i^k-\boldsymbol{I}_d\right)\boldsymbol{DLD}\left(\boldsymbol{S}_i^k-\boldsymbol{I}_d\right)\boldsymbol{L}_i^{\frac{1}{2}}\right]\right)\right\}.\tag{71}$$

In general cases, there is no easy way to find a optimal stepsize matrix $\boldsymbol{D}$ satisfying (70), alternatively, we choose the optimal diagonal stepsize $\boldsymbol{D}_3^*$ in this case similarly to [LKR23]. The stepsize condition for MARINA has already been described by (66). Note that we only consider MARINA, but not DIANA or DCGD-SHIFT, because DIANA and DCGD-SHIFT offer suboptimal rates compared to MARINA in the non-convex setting. For det-MARINA, we fix $\boldsymbol{W}=\boldsymbol{L}^{-1}$, and use $\boldsymbol{D}^*_{\boldsymbol{L}^{-1}}$ as the stepsize matrix. In theory, det-MARINA in this case should always out perform MARINA in terms of iteration complexity.

In Figure 2, in each plot, we observe that det-MARINA outperforms MARINA and the rest of the non-variance reduced methods. This is expected, since our theory confirms that det-MARINA indeed has a better rate compared to MARINA, and the stepsizes of the non-variance reduced methods are negatively affected by the neighborhood. When p is reasonably large, the variance reduced methods considered here outperform the non-variance reduced methods. In this experiment we consider only the comparison involving det-CGD.

I.4 Improvements on det-CGD

In this section, we compare det-CGD in the distributed case with det-MARINA, which are both algorithms using matrix stepsizes and matrix smoothness. The purpose of this experiment is to show that det-MARINA improves on the current state of the art matrix stepsize compressed gradient method when the objective function is non-convex. Throughout the experiment, $\lambda=0.3$ is fixed, and for det-CGD, $\varepsilon^2=0.01$ is fixed in order to determine its stepsize. For a thorough comparison, we select the stepsize for det-CGD in the following way. Let us denote the stepsize as $\boldsymbol{D}=\gamma_{\boldsymbol{W}}\cdot\boldsymbol{W}$, where $\gamma_{\boldsymbol{W}}\in\mathbb{R}_{++},\boldsymbol{W}\in\mathbb{S}^d_{++}$. We first fix a matrix $\boldsymbol{W}$, in this case, we pick $\boldsymbol{W}$ from the set $\{\boldsymbol{L}^{-1},\operatorname{diag}^{-1}(\boldsymbol{L}),\boldsymbol{I}_d\}$, and then we determine the optimal scaling $\gamma_{\boldsymbol{W}}$ for each case using the condition given in [LKR23] (see (70) and (71)). Then, we denote the matrix stepsizes for det-CGD

$$\boldsymbol{D}_1=\gamma_{\boldsymbol{I}_d}\cdot\boldsymbol{I}_d,\qquad\boldsymbol{D}_2=\gamma_{\operatorname{diag}^{-1}(\boldsymbol{L})}\cdot\operatorname{diag}^{-1}(\boldsymbol{L}),\qquad\boldsymbol{D}_3=\gamma_{\boldsymbol{L}^{-1}}\cdot\boldsymbol{L}^{-1}.\tag{72}$$

For det-MARINA, we use the stepsize $\boldsymbol{D}^*_{\boldsymbol{L}^{-1}}$, which is described in (67). In this experiment, we compare det-CGD using three stepsizes $\boldsymbol{D}_1,\boldsymbol{D}_2,\boldsymbol{D}_3$ with det-MARINA using stepsize $\boldsymbol{D}^*_{\boldsymbol{L}^{-1}}$.

From Figure 3, it is clear that det-MARINA outperforms det-CGD with all matrix optimal stepsizes with respect to a fixed $\boldsymbol{W}$ considered here. This is expected, since the convergence rate of non-variance reduced methods are affected by its neighborhood. This experiment demonstrates the advantages of det-MARINA over det-CGD, and is also supported by our theory. Notice that though different $\boldsymbol{W}$ are considered for det-CGD, their convergence rates are similar, which is also mentioned by [LKR23].

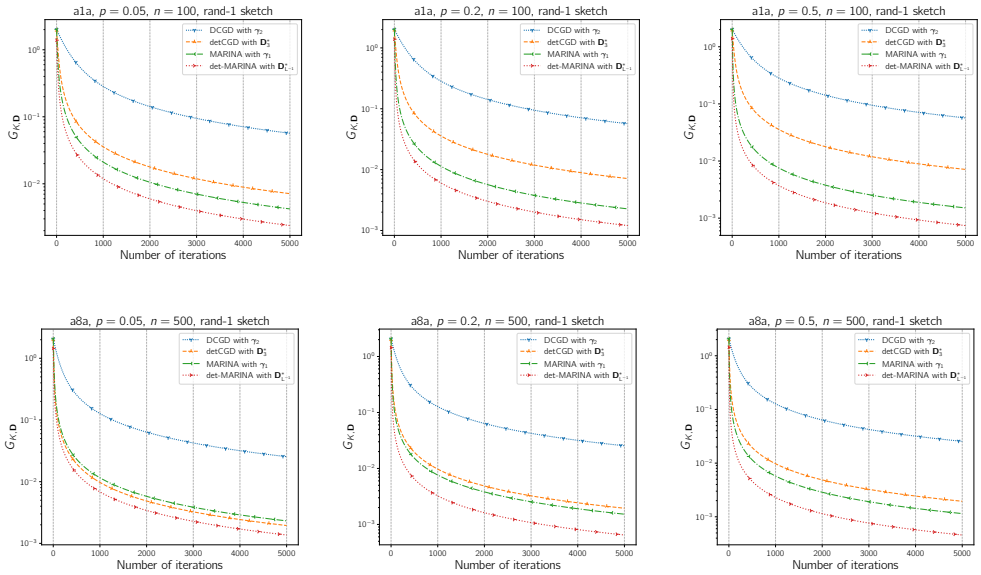

Figure 2: Comparison of DCGD with optimal scalar stepsize γ_2, det-CGD with optimal diagonal stepsize $\boldsymbol{D}_3^*$, MARINA with optimal scalar stepsize γ_1, and det-MARINA with optimal stepsize $\boldsymbol{D}_{\boldsymbol{L}^{-1}}^*$ with respect to $\boldsymbol{W} = \boldsymbol{L}^{-1}$. In each case, probability p is chosen the set $\{0.05, 0.2, 0.5\}$ for MARINA and det-MARINA. $\lambda = 0.3$ is fixed throughout the experiment. The notation n in the title indicates the number of clients in each case.

I.5 Comparing Different Stepsize Choices

This experiment is designed to see the how det-MARINA works under different stepsize choices. As it is mentioned in Appendix I.2, for each choice of $\boldsymbol{W} \in \mathbb{S}_{++}^d$, an optimal stepsize $\boldsymbol{D}_{\boldsymbol{W}}^*$ can be determined. Here we compare det-MARINA using three different stepsize choices $\boldsymbol{D}_{\boldsymbol{L}^{-1}}^*, \boldsymbol{D}_{\mathrm{diag}^{-1}(\boldsymbol{L})}^*$ and $\boldsymbol{D}_{\boldsymbol{I}_d}^*$. There stepsizes are explicitly defined in (67). Throughout the experiment, we fix $\lambda = 0.3$, rand-1 sketch is used in all cases.

We can observe from Figure 4 that, in almost all cases det-MARINA with stepsize $\boldsymbol{D}_{\mathrm{diag}^{-1}(\boldsymbol{L})}^*$ and $\boldsymbol{D}_{\boldsymbol{L}^{-1}}^*$ outperforms det-MARINA with $\boldsymbol{D}_{\boldsymbol{I}_d}^*$. As det-MARINA with $\boldsymbol{D}_{\boldsymbol{I}_d}^*$ can be viewed as MARINA using scalar stepsize but under matrix Lipschitz gradient assumption, this demonstrates the effectiveness of using a matrix stepsize over the scalar stepsize. However, in Figure 4, there are cases where det-MARINA with $\boldsymbol{D}_{\mathrm{diag}^{-1}(\boldsymbol{L})}^*$ outperforms $\boldsymbol{D}_{\boldsymbol{L}^{-1}}^*$. This tells us the two stepsizes are perhaps incomparable in general cases. This is similar to det-CGD, where optimal stepsizes with respect to a subspace associated with a fixed $\boldsymbol{W}^{-1}$ are incomparable.

I.6 Comparing Communication Complexity

In this section, we perform an experiment on how different probabilities p will affect the overall communication complexity of det-MARINA. We use $\boldsymbol{D}_{\boldsymbol{L}^{-1}}^*$ as the stepsize, which is determined with respect to the sketch used. Rand-τ sketches are used in these experiments, and we vary the minibatch size τ to provide a more comprehensive comparison. For rand-τ sketch $\boldsymbol{S}$ and any $\boldsymbol{A} \in \mathbb{S}_{++}^d$, one can show that

$$\mathbb{E}\left[\boldsymbol{S}\boldsymbol{A}\boldsymbol{S}\right] = \frac{d}{\tau}\left(\frac{d-\tau}{\mathrm{d}-1}\,\mathrm{diag}(\boldsymbol{A}) + \frac{\tau-1}{d-1}\boldsymbol{A}\right). \tag{73}$$

Combining (73) and (67), we can find out the corresponding matrix stepsize easily. In the experiment, a fixed number of iterations ($K = 5000$) is performed for each det-MARINA with the corresponding stepsize.

As it can be observed from Figure 5, in each dataset, the communication complexity tends to increase with the increase of probability p. However, when the number of iteration is fixed, a larger p often

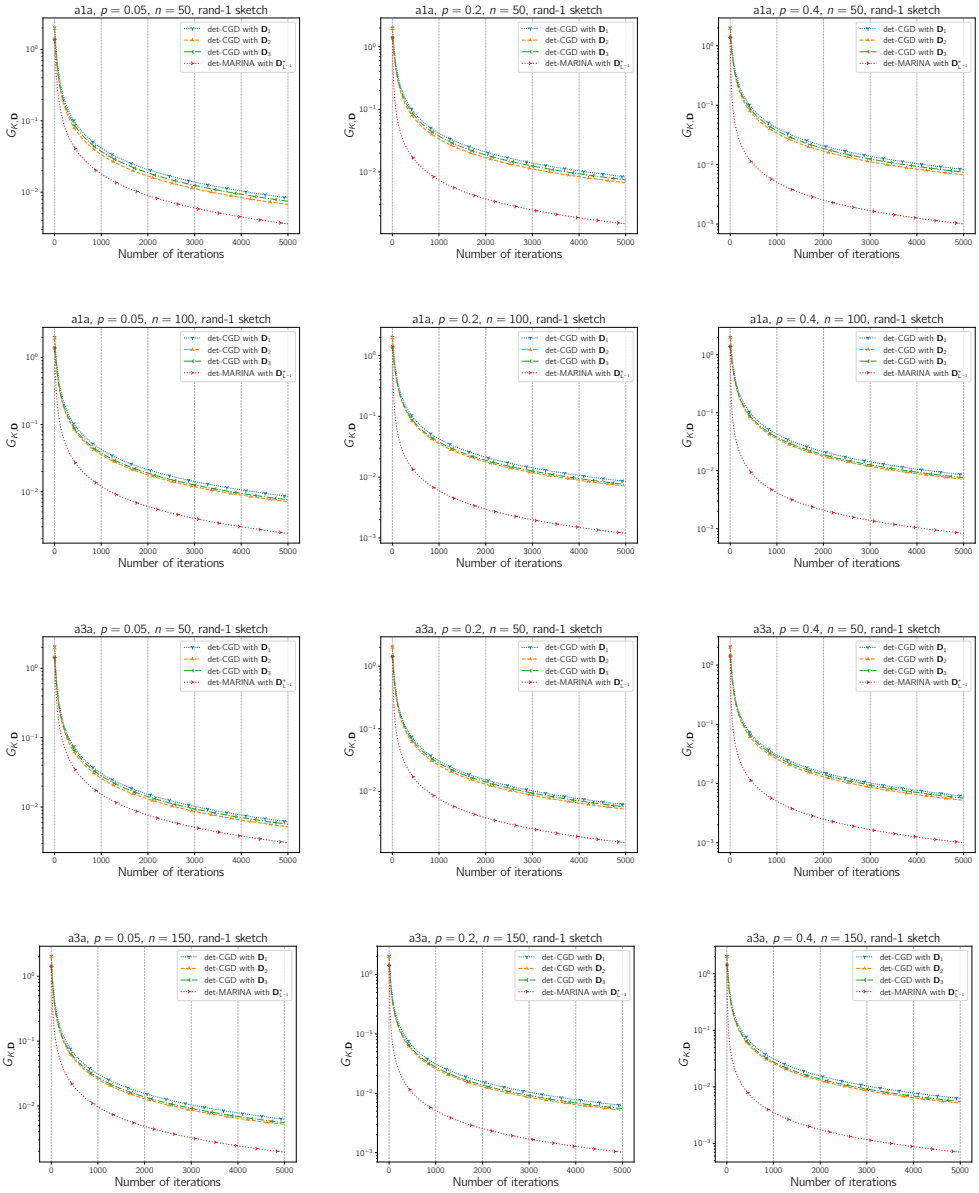

Figure 3: Comparison of `det-CGD` with matrix stepsize $\boldsymbol{D}_1$, $\boldsymbol{D}_2$ and $\boldsymbol{D}_3$ and `det-MARINA` with optimal matrix stepsize with respect to $\boldsymbol{W} = \boldsymbol{L}^{-1}$. The stepsizes $\{\boldsymbol{D}_i\}_{i=1}^3$ are described in (72). Throughout the experiment ε^2 is fixed at 0.01, the notation p in the title refers to the probability for `det-MARINA`, n denotes the number of clients considered, rand-1 sketch is used in all cases for all the algorithms.

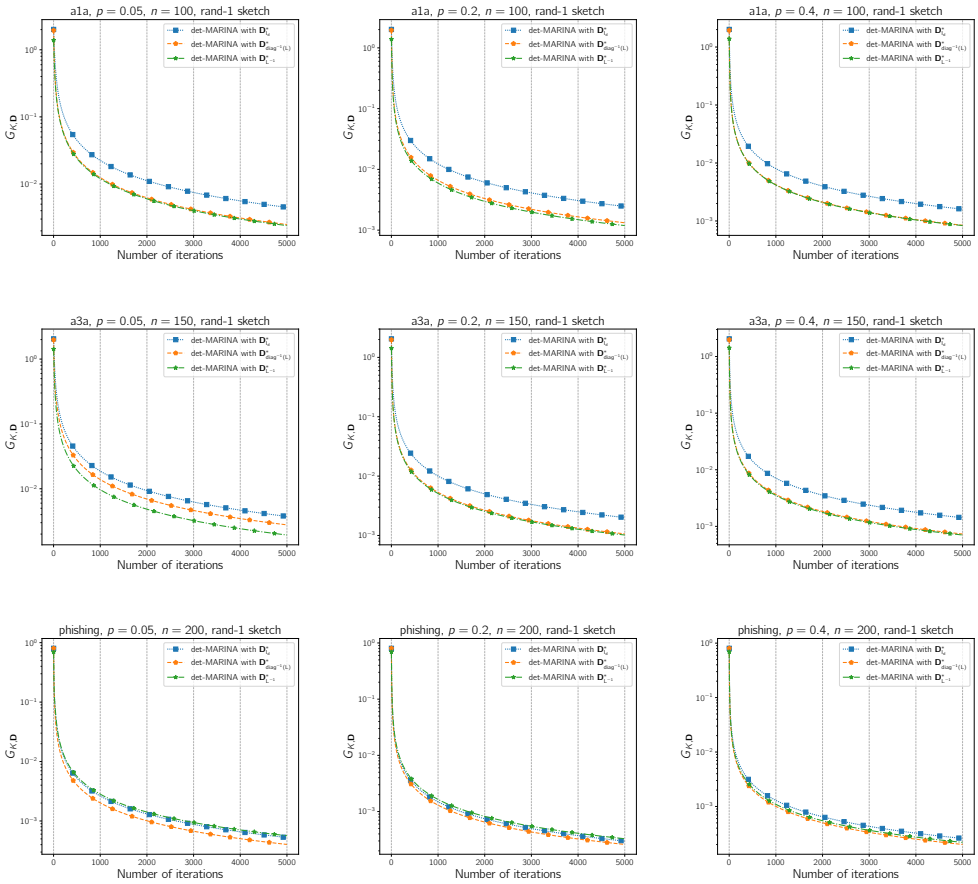

Figure 4: Comparison of `det-MARINA` with matrix stepsize $\boldsymbol{D}^*_{\boldsymbol{I}_d}$, $\boldsymbol{D}^*_{\text{diag}^{-1}(\boldsymbol{L})}$ and $\boldsymbol{D}^*_{\boldsymbol{L}^{-1}}$. The stepsizes are defined in (67). Throughout the experiment, $\lambda = 0.3$ is fixed, rand-1 sketch is used in all cases. The notation p in the title indicates the probability of sending the true gradient for `det-MARINA`, n denotes the number of clients considered.

means a faster rate of convergence. This difference in communication complexity is more obvious when we are using the Rand-1 sketch. In real federated learning settings, there is often constraints on network bandwidth from clients to the server. Thus, trading off between communication complexity and iteration complexity, i.e. selecting the compression mechanism carefully to guarantee a acceptable speed that satisfies the bandwidth constraints, becomes important.

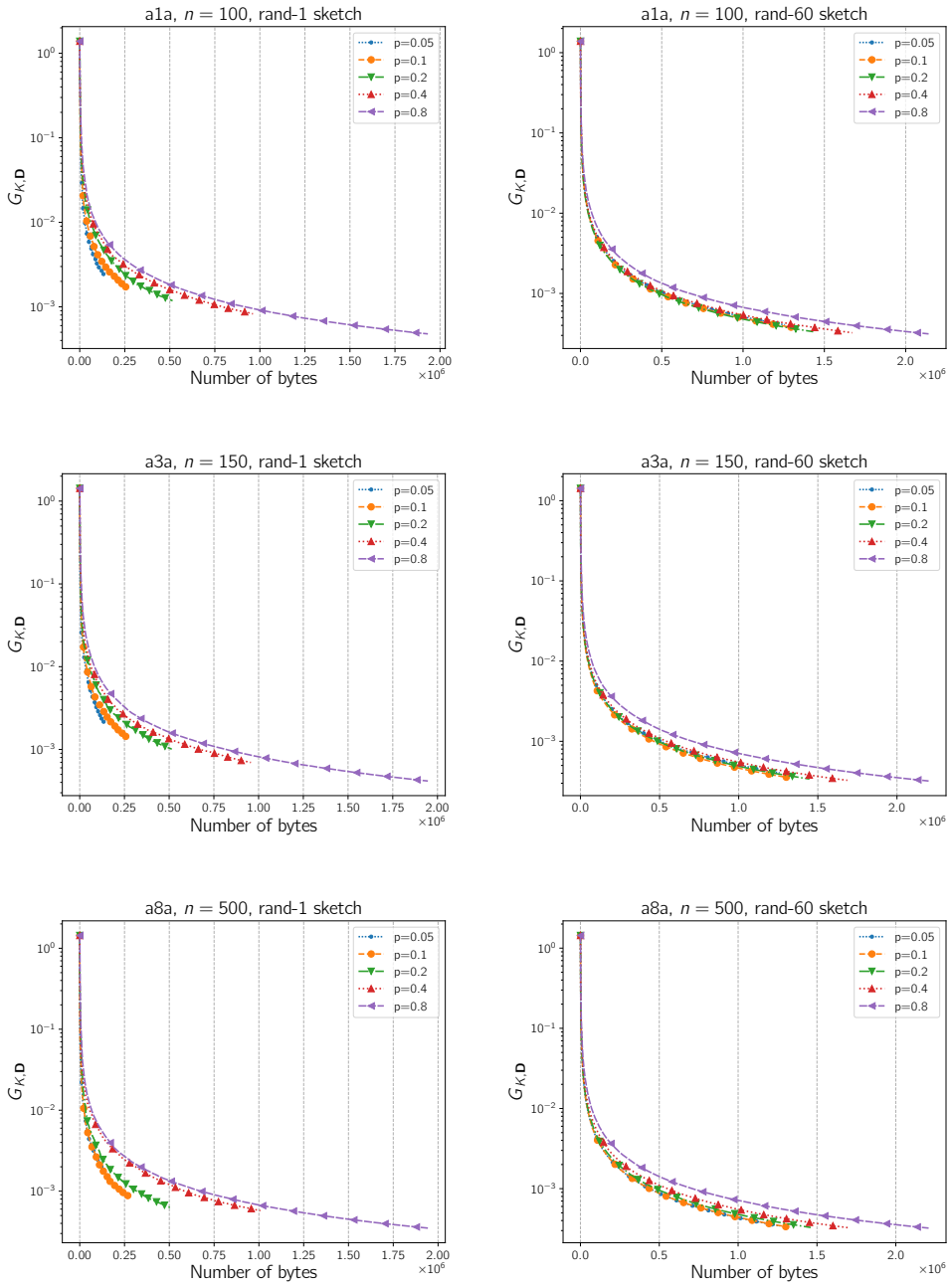

Figure 5: Comparison of `det-MARINA` with stepsize $\boldsymbol{D}^*_{\boldsymbol{L}^{-1}}$ using different probability p. The probability p here is chosen from the set $\{0.05, 0.1, 0.2, 0.4, 0.8\}$. The notation n in the title denote the number of clients considered. The x-axis is now the number of bytes sent from a single node to the server. In each case, `det-MARINA` is run for a fixed number of iterations $K = 5000$.