(Distributed) Fractional Gradient Descent with Matrix Stepsizes for Non-Convex Optimisation

Alokendu Mazumder[1*], **Keshav Vyas**[2], **Punit Rathore**[1]

[1] Robert Bosch Center for Cyber Physical Systems, Indian Institute of Science, Bengaluru, India.
[2] Independent Researcher, India.

Abstract

Fractional derivatives generalise integer-order derivatives, making them relevant for studying their convergence in descent-based optimisation algorithms. However, existing convergence analysis of fractional gradient descent is limited in both methods and settings. This paper bridges these gaps by establishing convergence guarantees for fractional gradient descent on a broader class of non-convex functions, known as matrix-smooth functions. We leverage the matrix smoothness properties of the function to prove convergence and accelerate the fractional gradient descent iterates. We propose two novel stochastic fractional descent algorithms, named *Compressed Fractional Gradient Descent* (CFGD), incorporating a matrix-valued stepsize to minimise matrix-smooth non-convex objectives. Our theoretical analysis covers both single-node and distributed settings and shows that matrix stepsizes better capture the structure of the objective, leading to faster convergence than scalar stepsizes. Additionally, we highlight the importance of matrix stepsizes to leverage model structure effectively. To the best of our knowledge, this is the first work to introduce fractional gradient descent in a federated/distributed setting.

1 Introduction

Minimising smooth and non-convex functions is a fundamental challenge in applied mathematics, with broad applications across various domains. Many machine learning algorithms rely on solving optimisation problems for both training and inference, often involving structural constraints or non-convex objectives to accurately address high-dimensional or non-linear tasks. However, non-convex problems are generally NP-hard, leading to the common strategy of relaxing them into convex problems and applying traditional optimisation techniques.

Despite the promise shown by direct approaches to non-convex optimisation, their convergence properties remain poorly understood, posing challenges for large-scale applications. While convex optimisation is more extensively studied and easier to solve, the non-convex setting is of greater practical relevance, often becoming the primary computational bottleneck in real-world problems.

In this paper, we consider the general minimisation problem:

$$\min_{x\in\mathbb{R}^d} f(x), \tag{1}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is a differentiable function. For this problem to have a finite solution we will assume throughout the paper that f is bounded from below.

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

The *stochastic gradient descent* (SGD) (Moulines and Bach 2011; Bubeck et al. 2015; Gower et al. 2019) algorithm is one of the most common algorithms to solve this problem. In its most general form, it can be written as

$$x_{t+1} = x_t - \alpha g_t \tag{2}$$

where g_t is the unbiased stochastic estimator of $\nabla f(x_t)$ and $\alpha > 0$ is a positive scalar stepsize. A particular case of interest is the *compressed gradient descent* (CGD) algorithm (Khirirat, Feyzmahdavian, and Johansson 2018), where the estimator g_t is taken as a compressed alternative of the initial gradient:

$$g_t = \mathcal{C}(\nabla f(x_t)). \tag{3}$$

The compressor $\mathcal{C}$ is designed as a sparse estimator to minimise communication overhead in distributed and federated environments. This compressor is stochastic, thereby introducing stochasticity into the fractional gradient descent iterates. This compressor reduces communication costs, addressing a major bottleneck in distributed optimisation (Konečnỳ 2016). Given the resource constraints of modern devices, various practical compression strategies are utilized, including compressing model updates from the server to clients and reducing computational loads during local training. While these strategies are complementary, gradient compression offers the most significant practical benefit (Kairouz et al. 2021), primarily due to slower client upload speeds and the advantages of gradient averaging. This work focuses on compressing fractional gradients. The stochasticity in fractional gradient descent in this work is introduced by the compressor. Thus, it can be considered a particular case of stochastic fractional gradient descent.

[*] Alokendu Mazumder is supported by the Kotak AI-ML Center PhD Fellowship and the Prime Minister's Research Fellowship, India.

Accepted at the 1^{st} Workshop on Federated Learning for Critical Applications (FLCA), in conjunction with AAAI 2026, Singapore.

An important subclass of compressors is sketches. Sketches are linear operators defined on $\mathbb{R}^d$, represented as $\mathcal{C}(w) = \mathbf{S}w$ for any $w \in \mathbb{R}^d$, where $\mathbf{S}$ is a random matrix. A common example is the Rand-k compressor, which randomly selects k entries from its input and scales them with a scalar multiplier to ensure the estimator remains unbiased. Instead of transmitting all d coordinates of the gradient, only a subset of size k is communicated, reducing the communication cost by a factor of d/k. Formally, Rand-k is defined as $\mathbf{S} = \sum_{j=1}^{k} \frac{d}{k} e_{i_j} e_{i_j}^{\top}$, where i_j denotes the selected coordinates of the input vector. For a detailed overview of compression techniques, refer to (Safaryan, Shulgin, and Richtárik 2022).

A particularly interesting development in optimisation is the use of *fractional gradient descent* (FGD) (Wei et al. 2020; Shin, Darbon, and Karniadakis 2021), where fractional derivatives replace traditional integer-order derivatives. Fractional derivatives combine integer-order derivatives with fractional integrals, with integrals easily generalised via the Cauchy repeated integral formula. Unlike standard derivatives, fractional derivatives—studied extensively (David, Linares, and Pallone 2011; Oldham and Spanier 1974; Luchko 2023)—extend the concept of differentiation, offering a more flexible and nuanced mathematical tool.

The basic concept of a fractional derivative is a combination of derivatives of integer order and fractional integrals. The fractional derivative that will be studied in this paper is the Caputo derivative, which has nice analytic properties. The definition from (Shin, Darbon, and Karniadakis 2021) is as follows where Γ is the gamma function generalising the factorial.

Definition 1. *The Caputo derivative of $f : \mathbb{R} \to \mathbb{R}$ of order $\beta \in (0, 1)$ is $(n = \lceil \beta \rceil)$:*

$$D_b^{\beta} = \frac{(sgn(x-b))^{n-1}}{\Gamma(n-\beta)} \int_b^x \frac{f^n(\tau)}{|x-\tau|^{\beta-n+1}} d\tau. \tag{4}$$

Notice that it depends directly on the ordinary n^{th}-order derivative of f, basically you take the classical slope at every past time τ, weight it by the power-law kernel $(x-\tau)^{\beta-n+1}$, and integrate. Unlike a local derivative $f^n(x)$, the Caputo derivative "remembers" the entire history of $f^n(x)$, yielding a non-local, memory-driven operator. This generalisation has found applications across various fields, raising an intriguing question: *Can the same principles be applied to optimisation, just as integer-order derivatives are used in gradient descent?*

Experimental results suggest that fractional methods have potential advantages in optimisation compared to integer-order methods (Shin, Darbon, and Karniadakis 2021). With carefully chosen hyperparameters, these methods can significantly outperform standard gradient descent, indicating promising avenues for further research. This paper focuses on the Caputo derivative-based fractional derivatives, known for their favorable analytic properties. The *Adaptive Terminal Caputo Fractional Gradient Descent* (AT-CFGD) method (Shin, Darbon, and Karniadakis 2021) empirically outperforms standard gradient descent in convergence rate. Their experiments demonstrate that training neural networks with AT-CFGD achieves faster convergence and lower testing error than traditional methods. With specifically chosen hyperparameters, fractional gradient descent can outperform standard gradient descent, suggesting that further study on the application of fractional derivatives to optimisation has significant potential.

In addition to assuming that the function f is bounded from below, we also assume it to be $\mathbf{L}$-matrix smooth. This assumption allows us to leverage the full information encoded in both the smoothness matrix $\mathbf{L}$ and the stepsize matrix $\mathbf{D}$.

Assumption 2 (Matrix Smoothness)**.** *There exists $\mathbf{L} \in \mathcal{S}_+^d$ such that*

$$f(x) \leq f(w) + \langle \nabla f(w), w - x \rangle + \frac{1}{2} \langle \mathbf{L}(w-x), w-x \rangle \tag{5}$$

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

The concept of matrix smoothness, which generalises scalar smoothness, has proven to be highly effective for enhancing the training of supervised models. By leveraging smoothness matrices alongside novel communication sparsification strategies, it addresses the communication overhead in distributed optimisation (Safaryan, Shulgin, and Richtárik 2022; Li, Karagulyan, and Richtárik 2023a). This method was applied to three distributed optimisation algorithms in the convex setting, leading to substantial communication savings and consistently outperforming existing baselines. These results demonstrate the utility of incorporating matrix smoothness to enhance distributed optimisation techniques.

A particularly useful instance is when the smoothness matrix is block-diagonal, which is relevant in various applications, such as *neural networks* (NN). In this scenario, each block corresponds to a specific layer of the network, with the smoothness of the nodes within the j^{th} layer captured by a matrix $\mathbf{L}_j$. Unlike scalar smoothness, matrix smoothness emphasizes the similarity among certain entries while allowing for differences in others, reflecting the increasing complexity of information across layers while maintaining similarity among nodes within the same layer. This behaviour has been visually observed in previous studies (Yosinski et al. 2015; Zintgraf et al. 2017).

Another motivation for using a layer-dependent stepsize is rooted in physics. In nature, the propagation speed of light varies in media of different densities due to frequency variations. Similarly, different layers in neural networks carry different information, metric systems, and scaling. Thus, the stepsizes need to be chosen accordingly to achieve optimal convergence.

Building on the findings that demonstrate fractional gradient descent can achieve faster convergence than standard gradient descent with appropriate hyperparameters (Shin, Darbon, and Karniadakis 2021), we propose extending fractional gradient descent to distributed and federated settings and analysing its performance. To facilitate this extension, we first compress the fractional gradient and define g_t as:

$$g_t = \mathcal{C}(\partial_b^{\beta,\delta} f(x_t)), \tag{6}$$

where $\delta \in \mathbb{R}$ and $\partial_b^{\beta,\delta} f(x)$ is the fractional gradient (defined clearly in **Section** 1) of order $\beta \in (0, 1)$.

In particular, we propose two novel matrix stepsized CFGD algorithms and analyse their convergence properties for non-convex matrix-smooth functions in single node as well as in distributed settings. We empirically show that fractional gradient descent converges faster with matrix stepsizes. As mentioned earlier, we place special emphasis on the block-diagonal case. Our sketches are inspired by (Li, Karagulyan, and Richtárik 2023a).

To the best of our knowledge, this is the first work to extend fractional gradient descent to distributed settings and a broader class of functions, known as matrix-smooth functions.

Related Work

Many successful convex optimisation techniques have been adapted for use in the non-convex setting. Examples include adaptivity (Dvinskikh et al. 2019; Zhang et al. 2020), variance reduction (J Reddi et al. 2016; Zhang et al. 2020), and acceleration (Guminov et al. 2019). A key paper for our work is (Shin, Darbon, and Karniadakis 2021), which provides a unified analysis scheme for fractional gradient descent in non-convex scenarios, but is limited to deterministic settings. Comprehensive overviews of fractional gradient descent in non-convex optimisation are given in (Shin, Darbon, and Karniadakis 2021; Aggarwal 2024).

Matrix-Based Step Size Methods Newton's method is a classical example of a matrix stepsized method and has been widely used in optimisation (Gragg and Tapia 1974; Miel 1980; Yamamoto 1987). However, calculating the stepsize as the inverse Hessian at each iteration is computationally intensive. Quasi-Newton methods, such as those proposed by (Broyden 1965; Dennis and Moré 1977; Al-Baali and Khalfan 2007; Al-Baali, Spedicato, and Maggioni 2014), use simpler estimators to approximate the inverse Hessian. One example is the Newton-Star algorithm (Islamov, Qian, and Richtárik 2021), discussed further in **Section** 2.

Sketched gradient descent, introduced by (Gower and Richtárik 2015), employs unbiased compressors through a sketch-and-project approach and was initially analysed in the context of the linear feasibility problem. Subsequent work expanded this with a variance-reduced version (Hanzely, Mishchenko, and Richtárik 2018).

In neural networks, many studies have explored layer-wise optimisation of the training loss. For instance, (Zheng et al. 2019) propose using different scalar stepsizes for each layer, while (Yu et al. 2017; Ginsburg et al. 2019) suggest layer-wise normalization for Stochastic Normalized Gradient Descent. Additionally, layer-wise compression techniques for distributed settings have been investigated (Dutta et al. 2020; Wang, Safaryan, and Richtárik 2022).

Distributed Compressed Gradient Descent (DCGD) (Khirirat, Feyzmahdavian, and Johansson 2018) has seen many improvements, such as those in (Horvóth et al. 2022). *Federated* (FL) learning algorithms with unbiased compressors have also gained attention (Alistarh et al. 2017; Mishchenko et al. 2024; Gorbunov et al. 2021; Mishchenko et al. 2022; Maranjyan, Safaryan, and Richtárik 2022; Horváth et al. 2023) in recent years. Recently, the det-CGD algorithm (Li, Karagulyan, and Richtárik 2023a) 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 a weighted stationarity point under a convex condition for the symmetric and positive-definite matrix stepsize. In addition to DCGD, the well-known non-convex distributed learning algorithm MARINA (Gorbunov et al. 2021) was later extended with matrix-based step-sizes in det-MARINA (Li, Karagulyan, and Richtárik 2023b). det-MARINA can be seen as variance variance-reduced extension of det-CGD. Our work is motivated by the original det-CGD algorithm (Li, Karagulyan, and Richtárik 2023a), rather than its variance-reduced variant. In future work, we will develop a variance-reduced version of our proposed methods.

Takeaway 1: Techniques like leveraging layer-wise structures, using matrix-based stepsizes, or employing compression mechanisms have not yet been explored in the context of fractional gradient descent.

Fractional Gradient Descent Replacing the derivative in gradient descent with a fractional derivative does not guarantee convergence to the optimum. The convergence in fractional gradient descent depends significantly on the choice of terminal (Wei et al. 2020, 2017; Aggarwal 2024), b. Fixed b values can result in non-zero gradients at convergence points. To address this, methods have been proposed (Wei et al. 2020; Shin, Darbon, and Karniadakis 2021; Aggarwal 2024) that adjust the terminal or the derivative order to ensure convergence to the optimal point. Alternatively, some approaches generalise gradient flow by modifying the time derivative to a fractional derivative, avoiding terminal dependence issues (Hai and Rosenfeld 2021).

Fractional derivatives can be defined in various ways (David, Linares, and Pallone 2011), with the Caputo and Riemann-Liouville derivatives being the most common. Some works (Sheng et al. 2020) simplify by using a first-degree approximation of the fractional derivative, while others take convex combinations of fractional and integer derivatives (Khan et al. 2018). Extensions of fractional gradient descent, such as a fractional Adam optimiser, have been proposed but often rely on crude approximations (Shin, Darbon, and Karniadakis 2023). For a deeper understanding of the convergence analysis of fractional gradient-based methods, we refer readers to (Elnady et al. 2025).

Fractional gradient descent has been applied to machine learning, showing improved performance in training neural networks (Han and Dong 2023; Wang et al. 2017). It has also been used to train convolutional neural networks (Wang, He, and Zhu 2022; Sheng et al. 2020) and models like radial basis function neural networks (Khan et al. 2018) and finite impulse response models with missing data (Tang 2023).

However, much of the literature is limited to specific function types or lacks strong convergence guarantees. Comprehensive theoretical results are rare, with some exceptions (Hai and Rosenfeld 2021; Wang et al. 2017). Our goal is to develop a rigorous methodology to establish convergence results for fractional gradient descent in more general non-convex settings. Fractional derivatives, often defined via integration, fall within the broader framework of nonlocal calculus, explored in optimisation theory (Nagaraj 2020).

Takeaway 2: Fractional gradient descent has been studied primarily for specific function types (Shin, Darbon, and Karniadakis 2021; Aggarwal 2024) but has shown potential in neural network training (Wang et al. 2017; Sheng et al. 2020; Han and Dong 2023). Empirical results are limited, and we aim to provide a comprehensive theoretical and empirical analysis of fractional gradient descent for matrix smooth functions.

Contributions

Our paper contributes in the following ways:

1. We propose two novel matrix stepsize CFGD stochastic algorithms in **Section** 2, representing the first analysis of fractional gradient descent with fixed matrix stepsize for nonconvex optimisation. A unified theorem in **Section** 3 guarantees stationarity for minimising matrix-smooth non-convex functions, showing our algorithms improve on the scalar stepsize alternatives.
2. We establish a general $\mathcal{O}(1/\sqrt{T})$ convergence result to a stationary point in all our theorems, showing that fractional gradient descent with well-chosen hyperparameters is more natural for optimising matrix smooth functions and empirically achieves faster convergence than integer order gradient-based algorithms.
3. Assuming less expensive server-to-client communication (Konečný 2016; Kairouz et al. 2021), we propose distributed versions of our algorithms in Section 4, following the standard FL scheme, and prove weighted stationarity guarantees. Our theorem recovers the result for *distributed CGD* (DCGD) in the scalar case and DCGD with matrix stepsize (Li, Karagulyan, and Richtárik 2023a) and improves it empirically.
4. We validate our theoretical results with experiments, with plots and framework provided in the *supplementary material.*

Preliminaries

Notations and Symbols Euclidean norm on $\mathbb{R}^d$ is defined as $\|.\|$. Bold capital letters denote matrices. $\mathbf{I}_d$ and $\mathbf{0}_d$ denotes $d \times d$ identity matrix and zero matrix respectively. Let $\mathcal{S}^d_{++}$ (resp. $\mathcal{S}^d_{+}$) be set of $d \times d$ symmetric positive definite (resp. semi-definite) matrices. Given $\mathbf{B} \in \mathcal{S}^d_{++}$ and $x \in \mathbb{R}^d$, we write $\|x\|_{\mathbf{B}} := \sqrt{\langle \mathbf{B}x, x\rangle}$, where $\langle ., . \rangle$ is the standard Euclidean inner product on $\mathbb{R}^d$. For matrix $\mathbf{B} \in \mathcal{S}^d_{++}$, we define by $\lambda_{max}(\mathbf{B})$ (resp. $\lambda_{min}(\mathbf{B})$) the largest (resp. smallest) eigenvalue of the matrix $\mathbf{B}$. Let $\mathbf{B}_j \in \mathbb{R}^{d_j \times d_j}$ and $d = \sum_{j=1}^{l} d_j$. The matrix $\mathbf{B} = \mathrm{Diag}(\mathbf{B}_1, \ldots, \mathbf{B}_l)$ is defined as a block diagonal $d \times d$ matrix where the j^{th} block is equal to $\mathbf{B}_j$. We will use $\mathrm{diag}(\mathbf{B}) \in \mathbb{R}^{d \times d}$ to denote the diagonal of any matrix $\mathbf{B} \in \mathbb{R}^{d \times d}$. Given a function $f : \mathbb{R}^d \to \mathbb{R}$. Its gradient and its Hessian at point $x \in \mathbb{R}^d$ are respectively denotes as $\nabla f(x)$ and $\nabla^2 f(x)$. For any matrix $\mathbf{M}$, $|\mathbf{M}|$ denotes its determinant. The i^{th} coordinate of a vector v is denoted by $v^{(i)}$.

Fractional Gradient Descent In this paper, we focus on a modified version of the AT-CFGD method proposed by (Shin, Darbon, and Karniadakis 2021), specifically from the standpoint of matrix smooth functions. The fractional gradient descent method is defined for a function $f : \mathbb{R}^d \to \mathbb{R}$ with parameters $\beta \in (0, 1)$ and $\delta \in \mathbb{R}$ as follows:

$$x_{t+1} = x_t - \alpha \partial^{\beta,\delta}_{b} f(x_t), \tag{7}$$

where $\partial^{\beta,\delta}_{b} f(x)$ can be written as:

$$\partial^{\beta,\delta}_{b} f(x) = \left[\partial^{\beta,\delta}_{b^{(1)}} f_{1,x}\left(x^{(1)}\right), \ldots, \partial^{\beta,\delta}_{b^{(d)}} f_{d,x}\left(x^{(d)}\right)\right]. \tag{8}$$

Here, $f_{k,x}(y) = f(x + (y - x^{(k)})e^{(k)})$ with $e^{(k)}$ is the unit vector in the k^{th} coordinate, $x^{(k)}$ is the k^{th} coordinate value of the vector x and $\partial^{\beta,\delta}_{b^{(k)}} f_{k,x}(y) = \frac{1}{D^{\beta}_{b^{(k)}} y}\left(D^{\beta}_{b^{(k)}} f_{k,x}(y) + \delta|y - b^{(k)}| D^{1+\beta}_{b^{(k)}} f_{k,x}(y)\right)$.

The intuition behind this method is based on Theorem 2.3 in (Shin, Darbon, and Karniadakis 2021), which explains that, for a Taylor expansion of $f_{k,x}$ around b, the term $\partial^{\beta,\delta}_{b^{(k)}} f(y)$ represents the derivative of a smoothed function. For $m \geq 2$, the m^{th} term is scaled by the coefficient $C_{m,\beta,\delta} = \left(\frac{\Gamma(2-\beta)\Gamma(m)}{\Gamma(m+1-\beta)} + \delta\frac{\Gamma(2-\beta)\Gamma(m)}{\Gamma(m-\beta)}\right)$. The value of δ influences the asymptotic behaviour of these coefficients *w.r.t* m. While the first term tends to zero as $m \to \infty$, the second term exhibits an asymptotic rate of $\delta(m - \beta)^{\beta}$, following Wendel's double inequality (Qi and Luo 2013).

To fully specify this method, it is crucial to determine how to select b. Several works in the literature estimate b by modelling it as a function of the iteration index t. For example, (Shin, Darbon, and Karniadakis 2021) recommend choosing $b_t = x_{t-z}$ for some positive integer z. In this work, we adopt the approach from (Aggarwal 2024), where each coordinate of b_t should satisfy

$$\left|x^{(i)}_t - b^{(i)}_t\right| = \mu_t \frac{df}{dx^{(i)}}(x_t) \quad \forall i \in [d], \tag{9}$$

with $\mu_t \in \mathbb{R} \quad \forall t$ carefully chosen. We set $\mu_t = -0.0675$ for all t in all of our experiments. Additionally, we set $\delta = 0$ in all our experiments to avoid the computation of higher-order gradients, as it is computationally expensive. In practice, the fractional gradient $\partial^{\beta,\delta}_{b_t} f(x_t)$, as given in (7), can be computed efficiently using Gauss-Jacobi quadrature, as detailed in (Shin, Darbon, and Karniadakis 2021).

2 Compressed Fractional Gradient Descent with Matrix Stepsize

Below we propose the two main algorithms (CFGD):

$$x_{t+1} = x_t - \mathbf{D}\mathbf{A}_t \partial^{\beta,\delta}_{b_t} f(x_t), \tag{CFGD-1}$$

and

$$x_{t+1} = x_t - \mathbf{B}_t \mathbf{D} \partial^{\beta,\delta}_{b_t} f(x_t). \tag{CFGD-2}$$

$\mathbf{D} \in \mathcal{S}^d_{++}$ is the fixed stepsize matrix. The sequences of random matrices $\mathbf{A}_t$ and $\mathbf{B}_t$ are assumed to satisfy the following:

Assumption 3. *The random matrices (sketches) that appear in our proposed algorithms are i.i.d., unbiased, symmetric and positive semi-definite. Mathematically,*

$$\mathbf{A}_t, \mathbf{B}_t \in \mathcal{S}^d_+, \quad \mathbf{A}_t \overset{i.i.d.}{\sim} \mathcal{A} \quad \text{and} \quad \mathbf{B}_t \overset{i.i.d.}{\sim} \mathcal{B}$$
$$\mathbb{E}\left[\mathbf{A}_t\right] = \mathbb{E}\left[\mathbf{B}_t\right] = \mathbf{I}_d, \quad \text{for every} \quad t \in \mathbb{N}.$$

One can see that fractional GD is a special case of CFGD-1 and CFGD-2. Indeed, if $\mathbf{A}_t = \mathbf{B}_t = \mathbf{I}_d$ and $\mathbf{D} = \gamma \mathbf{I}_d$, then $x_{t+1} = x_t - \gamma \partial^{\beta,\delta}_{b_t} f(x_t)$.

Newton Star (NS) method (Islamov, Qian, and Richtárik 2021) demonstrated that under convexity assumptions, it achieves local quadratic convergence to the unique solution x^*. The update rule for the NS method is given by:

$$x_{t+1} = x_t - \left(\nabla^2 f(x^*)\right)^{-1} \nabla f(x_t). \tag{NS}$$

The NS method, despite its impressive convergence properties, is largely impractical in real-world applications due to its reliance on the Hessian matrix evaluated at the optimal point x^*. Accessing this matrix is typically infeasible in most practical scenarios. However, the method underscores an important idea: employing a constant matrix stepsize has the potential to enhance the convergence speed of gradient-based algorithms. This observation opens a promising avenue for further investigation, particularly in the context of fractional gradient descent algorithms, where such an analysis remains underexplored.

The update rules for Algorithms CFGD-1 and CFGD-2 differ in the sequence in which the sketch and stepsize are applied. Notably, these two algorithms become identical when the matrix multiplication of the sketch and stepsize is commutative. In fact, a straightforward relationship shows that by setting

$$\mathbf{B}_t = \mathbf{D}\mathbf{A}_t\mathbf{D}^{-1}, \tag{10}$$

the updates in CFGD-1 and CFGD-2 become identical. By defining $\mathbf{B}_t$ as in (10), we recover the unbiasedness condition $\mathbb{E}\left[\mathbf{B}_t\right] = \mathbf{D}\mathbb{E}\left[\mathbf{A}_t\right]\mathbf{D}^{-1} = \mathbf{I}_d$. However, in general $\mathbf{D}\mathbb{E}\left[\mathbf{A}_t\right]\mathbf{D}^{-1}$ is not necessarily symmetric, which contradicts to **Assumption** 3. Hence, CFGD-1 and CFGD-2 are not equivalent for our purposes.

3 Convergence Analysis of CFGD

Before starting the main results, we present a stepsize condition for CFGD-1 and CFGD-2

$$\mathbb{E}\left[\mathbf{A}_t\mathbf{D}\mathbf{L}\mathbf{D}\mathbf{A}_t\right] \preceq \mathbf{D}, \tag{11}$$

and

$$\mathbb{E}\left[\mathbf{D}\mathbf{B}_t\mathbf{L}\mathbf{B}_t\mathbf{D}\right] \preceq \mathbf{D}. \tag{12}$$

In vanilla, usually $\gamma < L^{-1}$ is standard condition for convergence. The above equations can be considered as matrix counterparts to stepsizes. We begin by defining an important proposition that is essential for understanding the main theorems. The below proposition is a special case of Lemma 19 of (Aggarwal 2024).

Proposition 1. *If $f : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable and $\boldsymbol{L}$ smooth, $\beta \in (0,1]$, then*

$$|\nabla^{(i)} f(x) - \partial^{\beta,\delta}_{b^{(i)}} f(x)| \leq K|x^{(i)} - b^{(i)}| \quad \forall i \in [d], \tag{13}$$

where $K = \frac{\lambda_{max}(\boldsymbol{L})(1-\beta)}{(2-\beta)}$

Below is the main convergence theorem for both algorithms in the single-node regime.

Theorem 1. *Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable, $\boldsymbol{L}$-smooth, and satisfies Assumptions 1 – 3. Let $\beta \in (0,1)$. Define K as in Proposition 1. Then, for each $t \geq 0$*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(x_t)\|^2_{\boldsymbol{D}}\right] \leq \frac{(f(x_0) - f^*)}{cT} \tag{14}$$

if one of the below conditions is true:

- *The vectors x_t and b_t are the iterates of CFGD-1 and (9) respectively, $\mu \in \left(\frac{-4.236}{K}, \frac{0.236}{K}\right)$ and $\boldsymbol{D}$ satisfies (11);*
- *The vectors x_t and b_t are the iterates of CFGD-2 and (9) respectively, $\mu \in \left(\frac{-4.236}{K}, \frac{0.236}{K}\right)$ and $\boldsymbol{D}$ satisfies (12).*

where $c = \left\{(1 - K\mu) - \frac{(1+K\mu)^2}{2}\right\}$

Notably, Theorem 1 provides the same convergence rate for any $\mathbf{D} \in \mathcal{S}^d_{++}$, even though the matrix norms on the left-hand side are not directly comparable across different matrices. To make the right-hand side of (14) comparable, it is essential to normalize the matrix $\mathbf{D}$ used for measuring the gradient norm. Following the determinant normalization method proposed by (Li, Karagulyan, and Richtárik 2023a), we divide both sides of (14) by $|\mathbf{D}|^{1/d}$, resulting in the following form:

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(x_t)\|^2_{\frac{\mathbf{D}}{|\mathbf{D}|^{1/d}}}\right] \leq \frac{(f(x_0) - f^*)}{c|\mathbf{D}|^{1/d}T} \tag{15}$$

This normalization is meaningful because adjusting the matrix to $\frac{\mathbf{D}}{|\mathbf{D}|^{1/d}}$ ensures that its determinant is 1, allowing the norm on the left-hand side to be comparable to the standard Euclidean norm. Importantly, the volume of the normalized ellipsoid $\{x \in \mathbb{R}^d : \|x\|^2_{\mathbf{D}/|\mathbf{D}|^{1/d}} \leq 1\}$ is independent of the choice of $\mathbf{D} \in \mathcal{S}^d_{++}$. Consequently, the results in (14) remain comparable across different choices of $\mathbf{D}$, as the right-hand side of (14) reflects the volume of the ellipsoid containing the gradient.

Optimal Matrix Stepsize for CFGD-1 and CFGD-2

In this section, we discuss selecting the optimal matrix stepsize to minimise iteration complexity. Maximizing the determinant of $\mathbf{D}$ is key to reducing the gradient norm in (15), but each algorithm has a unique constraint on $\mathbf{D}$: (11) for CFGD-1 and (12) for CFGD-2. The maximisation is straightforward for CFGD-2 but more complex for CFGD-1, which we tackle first.

According to (15), the optimal $\mathbf{D}$ is defined as the solution of the following constrained optimisation problem:

$$\begin{aligned} &\textit{maximize} && \log|\mathbf{D}| \\ &\textit{subject to} && \mathbb{E}\left[\mathbf{A}_t\mathbf{D}\mathbf{L}\mathbf{D}\mathbf{A}_t\right] \preceq \mathbf{D}, \\ & && \mathbf{D} \in \mathcal{S}^d_{++}. \end{aligned} \tag{16}$$

Proposition 2. *The optimisation problem (16) w.r.t stepsize matrix $\boldsymbol{D} \in \mathcal{S}^d_{++}$, is a concave optimisation problem with convex set.*

This proposition is taken from (Li, Karagulyan, and Richtárik 2023a). One can easily convert the maximisation problem into minimisation and the objective function will become convex. Then it will be a convex optimisation problem with a convex set. The proof of this proposition is deferred to the *supplementary material* due to space constraints.

The CVXPY package (Diamond and Boyd 2016) could be used to solve (16), but only after transforming it into a *disciplined convex programming* (DCP) form as outlined by (Grant, Boyd, and Ye 2006). However, in general, (11) does not meet the DCP criteria. To enable the use of CVXPY, further modifications specific to this problem are required.

Finding the optimal stepsize for CFGD-2 is easier, in fact, we notice that (12). It is equivalent to

$$\mathbf{D} \preceq \left(\mathbb{E}\left[\mathbf{B}_t \mathbf{L} \mathbf{B}_t\right]\right)^{-1}. \tag{17}$$

On careful inspection, it is easy to identify that the map $g : \mathbf{B} \to \mathbf{BLB}$ is convex on $\mathcal{S}^d_{++}$. Hence, Jensen's inequality implies

$$\mathbb{E}\left[\mathbf{B}_t \mathbf{L} \mathbf{B}_t\right] \succeq \mathbb{E}\left[\mathbf{B}_t\right] \mathbf{L} \mathbb{E}\left[\mathbf{B}_t\right] \succeq \mathbf{L} \succ \mathbf{O}_d.$$

Since, both $\mathbf{D}$ and $\left(\mathbb{E}\left[\mathbf{B}_t \mathbf{L} \mathbf{B}_t\right]\right)^{-1}$ are positive definite, then the right-hand side of (15) is minimised exactly when

$$\mathbf{D} = \left(\mathbb{E}\left[\mathbf{B}_t \mathbf{L} \mathbf{B}_t\right]\right)^{-1} \preceq \mathbf{L}^{-1}. \tag{18}$$

4 Distributed Setting

In this section, we describe the distributed versions of our algorithms and present convergence guarantees for them. Let us consider an objective function that is sum decomposable:

$$f(x) := \frac{1}{n} \sum_{j=1}^{n} f_j(x),$$

where each $f_j : \mathbb{R}^d \to \mathbb{R}$ is a differentiable function. We assume that f satisfies Assumption 1 and the component functions satisfy the below condition.

Assumption 4. Each component function f_j is $\mathbf{L}_j$ smooth and is bounded from below: $f_j(x) \geq f_j^*$ for all $x \in \mathbb{R}^d$.

This assumption implies f has matrix smoothness with $\bar{\mathbf{L}} = \frac{1}{n}\sum_{j=1}^{n} \mathbf{L}_j \in \mathcal{S}^d_{++}$. In the standard federated learning setup (McMahan et al. 2016, 2017; Khirirat, Feyzmahdavian, and Johansson 2018), j^{th} client stores f_j, computes and compresses ∇f_j in parallel, and sends it to the central server. The server aggregates these gradients, updates the iterate, and broadcasts it to clients. See the pseudo-code below for details.

Theorem 2. *Let $f_j : \mathbb{R}^d \to \mathbb{R}$ satisfy Assumption 4 and let f satisfy Assumption 1 and 2 with smoothness matrix $\mathbf{L}$. Let $\beta \in (0, 1)$. Define $K_j \quad \forall j \in [n]$ as in Lemma 1. If the following conditions satisfy*

- $\boldsymbol{DLD} \preceq \boldsymbol{D}$
- $|x_t^{(i,j)} - b_t^{(i,j)}| = \mu_j \left|\frac{df}{dx^{(i,j)}}(x_t)\right|, \quad \mu_j \in \left(\frac{-4.236}{K_j}, \frac{0.236}{K_j}\right), \quad \forall i \in [d], j \in [n],$

then, for some $c, a > 0$, the following convergence bound is true for the iterates of Distributed CFGD-1 (Algorithm 1):

$$\min_{0 \leq t \leq T-1} \mathbb{E}\left[\|\nabla f(x_t)\|^2_{\frac{\boldsymbol{D}}{|\boldsymbol{D}|^{1/d}}}\right] \leq \frac{\left(1 + \frac{a^2 \lambda_{\boldsymbol{D}}}{n}\right)^T (f(x_0) - f^*)}{c|\boldsymbol{D}|^{1/d} T} + \frac{a^2 \lambda_{\boldsymbol{D}} \Delta^*}{c|\boldsymbol{D}|^{1/d} n}, \tag{19}$$

where $\Delta^* := f^* - \frac{1}{n}\sum_{j=1}^{n} f_j^*$,

$$\lambda_{\boldsymbol{D}} := \max_j \left\{\lambda_{max}\left(\mathbb{E}\left[\boldsymbol{L}_j^{\frac{1}{2}}\left(\boldsymbol{A}_{tj} - \boldsymbol{I}_d\right) \boldsymbol{DLD} \left(\boldsymbol{A}_{tj} - \boldsymbol{I}_d\right) \boldsymbol{L}_j^{\frac{1}{2}}\right]\right)\right\},$$

and $a^2 := \max_j (1 + K_j \mu_j)^2$.

The same conclusion holds for **Algorithm** 2 with a different constant $\lambda_{\mathbf{D}}$, as shown in the *supplementary material* along with the proofs for Theorem 2 and its counterpart for **Algorithm** 2, based on (Khaled and Richtárik 2020). Examining the right-hand side of (19), we observe an exponential dependence on K in the first term. However, $1 + a^2\lambda_{\mathbf{D}}/n$, influenced by the stepsize matrix $\mathbf{D}$, depends quadratically on $\mathbf{D}$. Thus, $\lambda_{\kappa\mathbf{D}} = \kappa^2 \lambda_{\mathbf{D}}$, rather than scaling linearly as in $|\kappa\mathbf{D}|^{1/d}$. By choosing a small coefficient κ, we ensure $\lambda_{\mathbf{D}}$ is of order n/K, allowing us to bound the numerator in the first term for a given K by selecting a sufficiently small matrix stepsize.

Corollary 1. *We reach an ϵ-stationarity, that is the right-hand side of (19) is upper bounded by ϵ^2, if the following conditions are satisfied:*

$$\boldsymbol{DLD} \preceq \boldsymbol{D}, \quad \lambda_{\boldsymbol{D}} \leq \min\left\{\frac{n}{T}, \frac{cn\epsilon^2}{2\Delta^*}|\boldsymbol{D}|^{1/d}\right\}, a^2 \leq 1, \quad T \geq \frac{6(f(x_0) - f^*)}{c|\boldsymbol{D}|^{1/d}\epsilon^2}. \tag{20}$$

One can easily see that the conditions on $\mathbf{D}$ are convex in nature. In order to minimise the iteration complexity for getting ϵ^2 error, one needs to solve the following optimisation problem

$$\begin{array}{ll} \textit{maximize} & \log|\mathbf{D}| \\ \textit{subject to} & \mathbf{D} \text{ satisfies (20).} \end{array} \tag{21}$$

Choosing the optimal stepsize for **Algorithm** 1 is analogous to solving (16). Furthermore, this leads to a convex matrix minimisation problem involving $\mathbf{D}$. Similar to the single-node case, computational methods can be employed using the CVXPY package. However, some additional effort is required to transform (20) into the *disciplined convex programming* (DCP) format.

The second term in (19) represents the convergence neighbourhood, which is independent of iteration count but depends on the number of clients n. Generally, this term, Δ^*/n, can grow unbounded as $n \to \infty$. However, by selecting a sufficiently small $\kappa > 0$, one can ensure $\lambda_{\mathbf{D}} \leq n/T$, allowing the neighbourhood term to approach zero as $T \to \infty$ with an appropriate stepsize choice. Corollary 1 summarizes these arguments; its proof can be found in the *supplementary material*.

Algorithm 1: Distributed (CFGD-1) (DCFGD-1)

```
Input: $x_0$, clients $n$, fractional order $\beta \in (0,1]$, stepsize matrix $\mathbf{D}$, iterations $T$
for $t = 0$ to $T-1$ do
    Devices ($j \in [n]$):
    sample $\mathbf{A}_{tj} \sim \mathcal{A}$
    compute $\mathbf{A}_{tj}\, \partial_{b_t}^{\beta,\delta} f_j(x_t)$
    broadcast $\mathbf{A}_{tj}\, \partial_{b_t}^{\beta,\delta} f_j(x_t)$
    Server:
    $g_t = \frac{1}{n}\mathbf{D} \sum_{j=1}^{n} \mathbf{A}_{tj}\, \partial_{b_t}^{\beta,\delta} f_j(x_t)$
    $x_{t+1} = x_t - g_t$
    broadcast $x_{t+1}$
end for
Return: $x_T$
```

Algorithm 2: Distributed (CFGD-2) (DCFGD-2)

```
Input: $x_0$, clients $n$, fractional order $\beta \in (0,1]$, stepsize matrix $\mathbf{D}$, iterations $T$
for $t = 0$ to $T-1$ do
    Devices ($j \in [n]$):
    sample $\mathbf{B}_{tj} \sim \mathcal{B}$
    compute $\mathbf{B}_{tj}\, \partial_{b_t}^{\beta,\delta} f_j(x_t)$
    broadcast $\mathbf{B}_{tj}\, \partial_{b_t}^{\beta,\delta} f_j(x_t)$
    Server:
    $g_t = \frac{1}{n} \sum_{j=1}^{n} \mathbf{B}_{tj} \mathbf{D}\, \partial_{b_t}^{\beta,\delta} f_j(x_t)$
    $x_{t+1} = x_t - g_t$
    broadcast $x_{t+1}$
end for
Return: $x_T$
```

Figure 1: Comparison of two distributed fractional gradient descent variants.

5 Experiments

In this section, we describe the settings and results of numerical experiments to demonstrate the effectiveness of our method. We perform several experiments under single node case and distributed case. Single node experiments can be found at **Section** E of the *supplementary material*.

Experimental Setup in Distributed Case

For the distributed case, we again use the logistic regression problem with a non-convex regulariser as our experiment setting. The objective is given as:

$$f(x) = \frac{1}{n}\sum_{j=1}^{n} f_j(x); \tag{22}$$

$$f_j(x) = \frac{1}{m_j}\sum_{p=1}^{m_j} \log\left(1 + e^{-b_{j,p}\cdot\langle a_{j,p}, x\rangle}\right) + \lambda \cdot \sum_{t=1}^{d} \frac{x_t^2}{1+x_t^2}; \tag{23}$$

where $x \in \mathbb{R}^d$ is the model, $(a_{j,p}, b_{j,p}) \in \mathbb{R}^d \times \{-1,+1\}$ is one data point in the dataset of client j whose size is m_j. $\lambda > 0$ is a constant associated with the regulariser. For each dataset used in the distributed setting, we randomly reshuffled the dataset before splitting it equally to each client. We estimate the smoothness matrices of function f and each individual function f_j here as:

$$\mathbf{L}_j = \frac{1}{m_j}\sum_{j=1}^{m_j} \frac{a_j a_j^\top}{4} + 2\lambda \cdot \mathbf{I}_d; \tag{24}$$

$$\mathbf{L} = \frac{1}{n}\sum_{j=1}^{n} \mathbf{L}_j. \tag{25}$$

The value of Δ^* here is determined in the following way, we first perform gradient descent on f and record the minimum value in the entire run, f^*, as the estimate of its global minimum, then we do the same procedure for each f_j to obtain the estimate of its global minimum f_j^*. After that, we estimate Δ^* using its definition.

Comparison with Standard DCGD and det-CGD

To ease the reading of this section we use DCFGD-1 (resp. DCFGD-2) to refer to Algorithm 1 (resp. Algorithm 2). This experiment is designed to show that DCFGD-1 and DCFGD-2 will have better iteration and communication complexity compared to standard DCGD (Khirirat, Feyzmahdavian, and Johansson 2018), DCGD with scalar stepsize and matrix smoothness, and det-CGD 1 (resp. det-CGD 2) (Li, Karagulyan, and Richtárik 2023a) with matrix stepsizes. We will use the standard DCGD here to refer to DCGD with a scalar stepsize and a scalar smoothness constant, and DCGD-mat to refer to the DCGD with a scalar stepsize with matrix smoothness. We also compare our proposed algorithm against det-MARINA (Li, Karagulyan, and Richtárik 2023b), the variance reduced version of det-DCGD. The Rand-1 sparsifier is used in all the algorithms throughout the experiment. The error level is fixed as $\epsilon^2 = 10^{-4}$, the conditions for the standard DCGD to converge can be deduced using Proposition 4 in (Khaled and Richtárik 2020), we use the largest possible scalar stepsize here for standard DCGD. The optimal scalar stepsize for DCGD-mat and optimal diagonal matrix stepsize $\mathbf{D}$ for det-DCGD 1, det-DCGD 2, DCFGD-1, and DCFGD-2, can be determined using Theorem 2. The optimal diagonal matrix stepsize for det-MARINA can be obtained from Corollary 4 of (Li, Karagulyan, and Richtárik 2023b). From the result of Figure 2, we can see that both DCFGD-1 and DCFGD-2 outperform standard DCGD and DCGD-mat in terms of iteration and communication complexity by a factor of more than 10^2, which confirms our theory. It also beats the state-of-the-art det-DCGD-1 (resp. det-DCGD-2) (Li, Karagulyan, and Richtárik 2023a) by a factor of 10 in iteration complexity. Notice that DCFGD-1, DCFGD-2 are expected to perform very similarly because the stepsize matrix and sketches are diagonal which means that they are commutable.

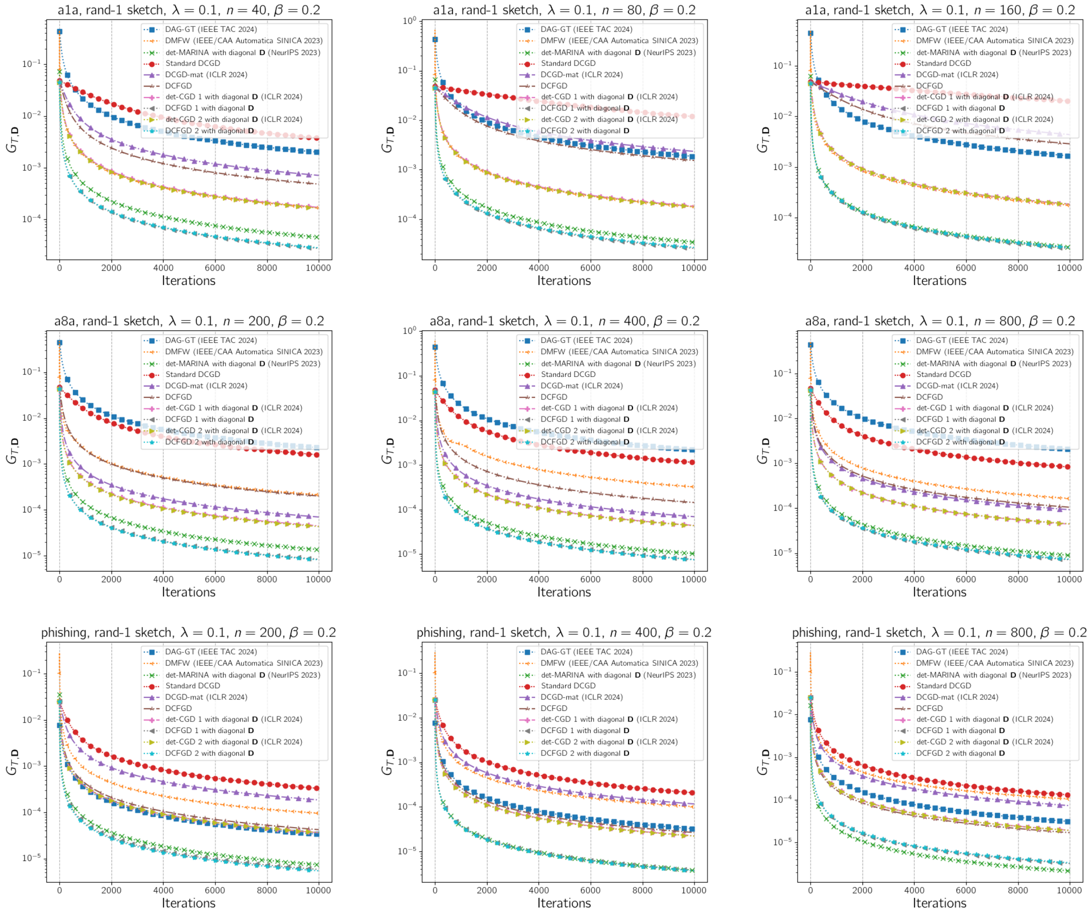


Figure 2: Comparison of standard DCGD, DCFGD, det-CGD (Li, Karagulyan, and Richtárik 2023a) with optimal diagonal stepsizes under rand-1 sketch, CFGD-1 (Ours) and CFGD-2 (Ours) with optimal diagonal stepsizes under rand-1 sketch, det-MARINA (Li, Karagulyan, and Richtárik 2023b), DAG-GT (Han et al. 2024), and DMFW (Hou et al. 2022). The stepsize for standard DCGD is determined from (det-DCGD) (Khaled and Richtárik 2020). Here $G_{T,\mathbf{D}} := \frac{1}{T}\sum_{t=0}^{T-1} \|\nabla f(x_t)\|^2_{\mathbf{D}/|\mathbf{D}|^{1/d}}$.

6 Conclusion

We propose two novel matrix stepsize-based fractional gradient descent algorithms that empirically demonstrate faster convergence compared to their scalar stepsize counterparts and traditional gradient descent. Additionally, we extend these algorithms to distributed and federated settings and show that they outperform well-known distributed algorithms, such as DCGD (Khirirat, Feyzmahdavian, and Johansson 2018) and det-CGD (Li, Karagulyan, and Richtárik 2023a), in both iteration complexity and communication efficiency. Our algorithms also perform competitively with the variance reduced det-MARINA.

Future Work

In this paper, we focused exclusively on linear sketches as the compression operator. However, many practical compressors do not belong to this category. Extending Algorithms CFGD-1 and CFGD-2 to accommodate general unbiased compressors presents an exciting direction for future research. Furthermore, inspired by the recent advancements in adaptive stepsizes (Loizou et al. 2021; Orvieto, Lacoste-Julien, and Loizou 2022; Schaipp, Gower, and Ulbrich 2023), developing an adaptive matrix stepsize specifically tailored to our framework could prove to be a promising direction.

Also, we plan to develop a variance reduced version of our proposed algorithm and benchmark it extensively against other variance reduction based stochastic optimisers.

Limitations

We achieve the same theoretical convergence rate as SGD because the fractional gradient is bounded in terms of the first-order gradient. However, we emphasize that the bound employed in this paper is more rigorous and technically sound compared to the approximations used for the fractional gradient in other works (Sheng et al. 2020; Khan et al. 2018).

References

Aggarwal, A. 2024. Convergence Analysis of Fractional Gradient Descent. *Transactions on Machine Learning Research*.

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

Al-Baali, M.; Spedicato, E.; and Maggioni, F. 2014. 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.

Alistarh, D.; Grubic, D.; Li, J.; Tomioka, R.; and Vojnovic, M. 2017. QSGD: Communication-efficient SGD via gradient quantization and encoding. *Advances in neural information processing systems*, 30.

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

Bubeck, S.; et al. 2015. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4): 231–357.

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

David, S. A.; Linares, J. L.; and Pallone, E. M. d. J. A. 2011. Fractional order calculus: historical apologia, basic concepts and some applications. *Revista Brasileira de Ensino de Física*, 33: 4302–4302.

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

Diamond, S.; and Boyd, S. 2016. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83): 1–5.

Dutta, A.; Bergou, E. H.; Abdelmoniem, A. M.; Ho, C.-Y.; Sahu, A. N.; Canini, M.; and Kalnis, P. 2020. On the discrepancy between the theoretical analysis and practical implementations of compressed communication for distributed deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3817–3824.

Dvinskikh, D.; Ogaltsov, A.; Gasnikov, A.; Dvurechensky, P.; Tyurin, A.; and Spokoiny, V. 2019. Adaptive gradient descent for convex and non-convex stochastic optimization. *arXiv preprint arXiv:1911.08380*.

Elnady, S. M.; El-Beltagy, M.; Radwan, A. G.; and Fouda, M. E. 2025. A comprehensive survey of fractional gradient descent methods and their convergence analysis. *Chaos, Solitons & Fractals*, 194: 116154.

Ginsburg, B.; Castonguay, P.; Hrinchuk, O.; Kuchaiev, O.; Lavrukhin, V.; Leary, R.; Li, J.; Nguyen, H.; Zhang, Y.; and Cohen, J. M. 2019. Stochastic gradient methods with layer-wise adaptive moments for training of deep networks. *arXiv preprint arXiv:1905.11286*.

Gorbunov, E.; Burlachenko, K. P.; Li, Z.; and Richtárik, P. 2021. MARINA: Faster non-convex distributed learning with compression. In *International Conference on Machine Learning*, 3788–3798. PMLR.

Gower, R. M.; Loizou, N.; Qian, X.; Sailanbayev, A.; Shulgin, E.; and Richtárik, P. 2019. SGD: General analysis and improved rates. In *International conference on machine learning*, 5200–5209. PMLR.

Gower, R. M.; and Richtárik, P. 2015. Randomized iterative methods for linear systems. *SIAM Journal on Matrix Analysis and Applications*, 36(4): 1660–1690.

Gragg, W. B.; and Tapia, R. A. 1974. Optimal error bounds for the Newton–Kantorovich theorem. *SIAM Journal on Numerical Analysis*, 11(1): 10–13.

Grant, M.; Boyd, S.; and Ye, Y. 2006. Disciplined Convex Programming Global Optimization: From Theory to Implementation ed L Liberti and N Maculan.

Guminov, S.; Nesterov, Y. E.; Dvurechensky, P.; and Gasnikov, A. 2019. Accelerated primal-dual gradient descent with linesearch for convex, nonconvex, and nonsmooth optimization problems. In *Doklady Mathematics*, volume 99, 125–128. Springer.

Hai, P. V.; and Rosenfeld, J. A. 2021. The gradient descent method from the perspective of fractional calculus. *Mathematical Methods in the Applied Sciences*, 44(7): 5520–5547.

Han, D.; Liu, K.; Lin, Y.; and Xia, Y. 2024. Distributed Adaptive Gradient Algorithm With Gradient Tracking for Stochastic Nonconvex Optimization. *IEEE Transactions on Automatic Control*, 69(9): 6333–6340.

Han, X.; and Dong, J. 2023. Applications of fractional gradient descent method with adaptive momentum in BP neural networks. *Applied Mathematics and Computation*, 448: 127944.

Hanzely, F.; Mishchenko, K.; and Richtárik, P. 2018. SEGA: Variance reduction via gradient sketching. *Advances in Neural Information Processing Systems*, 31.

Horváth, S.; Kovalev, D.; Mishchenko, K.; Richtárik, P.; and Stich, S. 2023. Stochastic distributed learning with gradient quantization and double-variance reduction. *Optimization Methods and Software*, 38(1): 91–106.

Horvóth, S.; Ho, C.-Y.; Horvath, L.; Sahu, A. N.; Canini, M.; and Richtárik, P. 2022. Natural compression for distributed deep learning. In *Mathematical and Scientific Machine Learning*, 129–141. PMLR.

Hou, J.; Zeng, X.; Wang, G.; Sun, J.; and Chen, J. 2022. Distributed momentum-based Frank-Wolfe algorithm for stochastic optimization. *IEEE/CAA Journal of Automatica Sinica*, 10(3): 685–699.

Islamov, R.; Qian, X.; and Richtárik, P. 2021. Distributed second order methods with fast rates and compressed communication. In *International conference on machine learning*, 4617–4628. PMLR.

J Reddi, S.; Sra, S.; Poczos, B.; and Smola, A. J. 2016. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. *Advances in neural information processing systems*, 29.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2021. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2): 1–210.

Khaled, A.; and Richtárik, P. 2020. Better theory for SGD in the nonconvex world. *arXiv preprint arXiv:2002.03329*.

Khan, S.; Naseem, I.; Malik, M. A.; Togneri, R.; and Bennamoun, M. 2018. A fractional gradient descent-based rbf neural network. *Circuits, Systems, and Signal Processing*, 37: 5311–5332.

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

Konečnỳ, J. 2016. Federated Learning: Strategies for Improving Communication Efficiency. *arXiv preprint arXiv:1610.05492*.

Li, H.; Karagulyan, A.; and Richtárik, P. 2023a. Det-CGD: Compressed gradient descent with matrix stepsizes for nonconvex optimization. *arXiv preprint arXiv:2305.12568*.

Li, H.; Karagulyan, A.; and Richtárik, P. 2023b. MARINA Meets Matrix Stepsizes: Variance Reduced Distributed Non-Convex Optimization. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*.

Loizou, N.; Vaswani, S.; Laradji, I. H.; and Lacoste-Julien, S. 2021. Stochastic polyak step-size for sgd: An adaptive learning rate for fast convergence. In *International Conference on Artificial Intelligence and Statistics*, 1306–1314. PMLR.

Luchko, Y. 2023. General fractional integrals and derivatives and their applications. *Physica D: Nonlinear Phenomena*, 133906.

Maranjyan, A.; Safaryan, M.; and Richtárik, P. 2022. Gradskip: Communication-accelerated local gradient methods with better computational complexity. *arXiv preprint arXiv:2210.16402*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

McMahan, H. B.; Yu, F.; Richtarik, P.; Suresh, A.; Bacon, D.; et al. 2016. Federated learning: Strategies for improving communication efficiency. In *Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain*, 5–10.

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

Mishchenko, K.; Gorbunov, E.; Takáč, M.; and Richtárik, P. 2024. Distributed learning with compressed gradient differences. *Optimization Methods and Software*, 1–16.

Mishchenko, K.; Malinovsky, G.; Stich, S.; and Richtárik, P. 2022. Proxskip: Yes! local gradient steps provably lead to communication acceleration! finally! In *International Conference on Machine Learning*, 15750–15769. PMLR.

Moulines, E.; and Bach, F. 2011. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Advances in neural information processing systems*, 24.

Nagaraj, S. 2020. Optimization and learning with nonlocal calculus. *arXiv preprint arXiv:2012.07013*.

Oldham, K.; and Spanier, J. 1974. *The fractional calculus theory and applications of differentiation and integration to arbitrary order*. Elsevier.

Orvieto, A.; Lacoste-Julien, S.; and Loizou, N. 2022. Dynamics of sgd with stochastic polyak stepsizes: Truly adaptive variants and convergence to exact solution. *Advances in Neural Information Processing Systems*, 35: 26943–26954.

Qi, F.; and Luo, Q.-M. 2013. Bounds for the ratio of two gamma functions: from Wendel's asymptotic relation to Elezović-Giordano-Pečarić's theorem. *Journal of Inequalities and Applications*, 2013: 1–20.

Safaryan, M.; Shulgin, E.; and Richtárik, P. 2022. 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.

Schaipp, F.; Gower, R. M.; and Ulbrich, M. 2023. A stochastic proximal polyak step size. *arXiv preprint arXiv:2301.04935*.

Sheng, D.; Wei, Y.; Chen, Y.; and Wang, Y. 2020. Convolutional neural networks with fractional order gradient method. *Neurocomputing*, 408: 42–50.

Shin, Y.; Darbon, J.; and Karniadakis, G. E. 2021. A Caputo fractional derivative-based algorithm for optimization. *arXiv preprint arXiv:2104.02259*.

Shin, Y.; Darbon, J.; and Karniadakis, G. E. 2023. Accelerating gradient descent and Adam via fractional gradients. *Neural Networks*, 161: 185–201.

Stich, S. U. 2019. Unified optimal analysis of the (stochastic) gradient method. *arXiv preprint arXiv:1907.04232*.

Tang, J. 2023. Fractional Gradient Descent-Based Auxiliary Model Algorithm for FIR Models with Missing Data. *Complexity*, 2023(1): 7527478.

Wang, B.; Safaryan, M.; and Richtárik, P. 2022. Theoretically better and numerically faster distributed optimization with smoothness-aware quantization techniques. *Advances in Neural Information Processing Systems*, 35: 9841–9852.

Wang, J.; Wen, Y.; Gou, Y.; Ye, Z.; and Chen, H. 2017. Fractional-order gradient descent learning of BP neural networks with Caputo derivative. *Neural networks*, 89: 19–30.

Wang, Y.; He, Y.; and Zhu, Z. 2022. Study on fast speed fractional order gradient descent method and its application in neural networks. *Neurocomputing*, 489: 366–376.

Wei, Y.; Chen, Y.; Cheng, S.; and Wang, Y. 2017. A note on short memory principle of fractional calculus. *Fractional Calculus and Applied Analysis*, 20(6): 1382–1404.

Wei, Y.; Kang, Y.; Yin, W.; and Wang, Y. 2020. Generalization of the gradient method with fractional order gradient direction. *Journal of the Franklin Institute*, 357(4): 2514–2532.

Yamamoto, T. 1987. A convergence theorem for Newton-like methods in Banach spaces. *Numerische Mathematik*, 51: 545–557.

Yosinski, J.; Clune, J.; Nguyen, A.; Fuchs, T.; and Lipson, H. 2015. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*.

Yu, A. W.; Huang, L.; Lin, Q.; Salakhutdinov, R.; and Carbonell, J. 2017. Block-normalized gradient method: An empirical study for training deep neural network. *arXiv preprint arXiv:1707.04822*.

Zhang, J.; Karimireddy, S. P.; Veit, A.; Kim, S.; Reddi, S.; Kumar, S.; and Sra, S. 2020. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33: 15383–15393.

Zheng, Q.; Tian, X.; Jiang, N.; and Yang, M. 2019. Layer-wise learning based stochastic gradient descent method for the optimization of deep convolutional neural network. *Journal of Intelligent & Fuzzy Systems*, 37(4): 5641–5654.

Zintgraf, L. M.; Cohen, T. S.; Adel, T.; and Welling, M. 2017. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*.

Supplementary Material

Contents

This *supplementary material* is part of the submission *(Distributed) Fractional Gradient Descent with Matrix Stepsizes for Non-Convex Optimisation*

A Single node case

Proof of Proposition 1

Proof sketch: **Lemma** 1 relates the first-order derivative with the $\beta \in (0, 1]$ order fractional derivative at each coordinate. We first start by relating $\mathbf{L}$-smoothness with scalar smoothness and then establish the inequality given in **Lemma** 1. Before proving the **Lemma** 1, we introduce some useful auxiliary lemmas.

Lemma 1. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuously differentiable function. If f is $\mathbf{L}$-smooth, where $\mathbf{L} \in \mathcal{S}_+^d$ (the set of $d \times d$ symmetric positive semidefinite matrices), then f is also $\lambda_{\max}(\boldsymbol{L})$-smooth, where $\lambda_{\max}(\boldsymbol{L})$ is the maximum eigenvalue value of $\boldsymbol{L}$.*

Proof. The proof is trivial. One can use the eigenvalue-inequality of the matrix $\mathbf{L}$ in (5). □

Now we will relate first-order gradient and fractional gradient coordinate-wise (*i.e. for every* $i \in [d]$). Before establishing a relationship, we present the following lemma:

Lemma 2. *If $f : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable and $\mathbf{L}$ smooth, then for each of its coordinate $i \in [d]$, $\nabla^{(i)} f(x)$ is $\lambda_{max}(\boldsymbol{L})$ continuous.*

Proof. From Lemma 1, f is $\lambda_{max}(\mathbf{L})$ smooth .

$$\begin{aligned} \|\nabla f(x) - \nabla f(y)\|_2 \leq & \quad \lambda_{max}(\mathbf{L})\|x - y\|_2 \\ \sqrt{\sum_{i=1}^{d} \left(\frac{df}{dx^{(i)}} - \frac{df}{dy^{(i)}} \right)^2} \leq & \quad \lambda_{max}(\mathbf{L})\|x - y\|_2 \end{aligned}$$

Since RHS is a real number and LHS is a sum of squares. The above inequality should hold individually for all $i \in [d]$. □

Lemma 3 ((Aggarwal 2024))**.** *Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable and $\mathbf{L}$ smooth. Let $\beta \in (0, 1]$ and $t \in \mathbb{R}^d$. Define $\phi_x^{(i)}(t)$ as:*

$$\phi_x^{(i)}(t) = f(t) - f(x) - \nabla^{(i)} f(x)(t^{(i)} - x^{(i)}).$$

Then, we have:

$$D^{\beta}_{b^{(i)}} f(x) - \frac{\nabla^{(i)} f(x)(x^{(i)} - b^{(i)})}{\Gamma(2-\beta)\left|x^{(i)} - b^{(i)}\right|} = \frac{-\phi_x^{(i)}(b)}{\Gamma(1-\beta)\left|x^{(i)} - b^{(i)}\right|} - \frac{\beta sign\left(x^{(i)} - b^{(i)}\right)}{\Gamma(1-\beta)} \int_{b^{(i)}}^{x^{(i)}} \frac{\phi_x^{(i)}(t)}{\left|x^{(i)} - t^{(i)}\right|^{\beta+1}} dt^{(i)}$$

for all $i \in [d]$.

Proof. Proof is via integration by parts following the logic of **Proposition** 3.1 of (Hai and Rosenfeld 2021).

As f is $\mathbf{L}$ smooth, it implies $\nabla^{(i)} f$ is $\lambda_{max}(\mathbf{L})$ continuous (using Lemma 2) for all $i \in [d]$. Note, for $\beta \in (0, 1]$, $D^{\beta}_{b^{(i)}} x^{(i)} = \frac{x^{(i)} - b^{(i)}}{\Gamma(2-\beta)\left|x^{(i)} - b^{(i)}\right|}$. Also, for the i^{th} coordinate, we have $d\phi_x^{(i)}(t) = (\nabla^{(i)} f(t) - \nabla^{(i)} f(x))dt^{(i)}$. Now, we begin with the following:

$$\begin{aligned} D^{\beta}_{b^{(i)}} f(x) - \nabla^{(i)} f(x) D^{\beta}_{b^{(i)}}\left(x^{(i)}\right) = & \quad \frac{1}{\Gamma(1-\beta)} \int_{b^{(i)}}^{x^{(i)}} \left|x^{(i)} - t^{(i)}\right|^{-\beta} (\nabla^{(i)} f(t) - \nabla^{(i)} f(x)) dt^{(i)} \\ = & \quad \frac{1}{\Gamma(1-\beta)} \int_{b^{(i)}}^{x^{(i)}} \left|x^{(i)} - t^{(i)}\right|^{-\beta} d\phi_x^{(i)}(t) \\ = & \quad \left[\frac{\left|x^{(i)} - t^{(i)}\right|^{-\beta} \phi_x^{(i)}(t)}{\Gamma(1-\beta)} \right]_{t^{(i)} = b^{(i)}}^{x^{(i)}} - \frac{\beta}{\Gamma(1-\beta)} \int_{b^{(i)}}^{x^{(i)}} \left|x^{(i)} - t^{(i)}\right|^{-\beta-1} sgn(x^{(i)} - t^{(i)}) \phi_x^{(i)}(t) dt^{(i)} \\ = & \quad \left[\frac{\phi_x^{(i)}(t)}{\Gamma(1-\beta)\left|x^{(i)} - t^{(i)}\right|^{\beta}} \right]_{t^{(i)} = b^{(i)}}^{x^{(i)}} - \frac{\beta sgn(x^{(i)} - t^{(i)})}{\Gamma(1-\beta)} \int_{b^{(i)}}^{x^{(i)}} \frac{\phi_x^{(i)}(t)}{\left|x^{(i)} - t^{(i)}\right|^{\beta+1}} dt^{(i)} \end{aligned}$$

The first term vanishes as $t^{(i)} \to x^{(i)}$. One can show this using L'Hopital's rule:

$$\lim_{x^{(i)} \to t^{(i)}} \frac{\phi_x^{(i)}(t)}{\Gamma(1-\beta)\left|x^{(i)} - t^{(i)}\right|^{\beta}} = \lim_{x^{(i)} \to t^{(i)}} \frac{\nabla^{(i)} f(t) - \nabla^{(i)} f(x)}{\beta\Gamma(1-\beta)\left|x^{(i)} - t^{(i)}\right|^{\beta-1} sgn(x^{(i)} - t^{(i)})} = 0$$

The last equality is due to $\beta - 1 \leq 0$. □

Lemma 4 (Relationship between first-order derivative and fractional derivative (Aggarwal 2024))**.** *Suppose $f : \mathbb{R} \to \mathbb{R}$ is continuously differentiable. Let $\beta \in (0, 1]$. If f is $\boldsymbol{L}$ smooth, then for each coordinate $i \in [d]$ it satisfies:*

$$\left| \frac{\nabla^{(i)} f(x) \left(x^{(i)} - b^{(i)}\right)}{\Gamma(2-\beta) \left|x^{(i)} - b^{(i)}\right|^{\beta}} - D^{\beta}_{b^{(i)}} f(x) \right| \le \frac{\lambda_{max}(\boldsymbol{L})}{\Gamma(1-\beta)(2-\beta)} \left|x^{(i)} - b^{(i)}\right|^{2-\beta}$$

Proof. Note that $\lambda_{\max}(\mathbf{L})$-smooth implies that $\phi^{(i)}_x(t) \le \frac{\lambda_{\max}(\mathbf{L})}{2}|x^{(i)} - t^{(i)}|^2$. Also, $2 - \beta > 0$ since $\beta \in (0, 1]$. Thus,

$$\begin{aligned}
\frac{\nabla^{(i)} f(x)(x^{(i)} - b^{(i)})}{\Gamma(2-\beta)|x^{(i)} - b^{(i)}|^{\beta}} - D^{\beta}_{b^{(i)}} f(x) &= \frac{\phi^{(i)}_x(b)}{\Gamma(1-\beta)|x^{(i)} - b^{(i)}|^{\beta}} + \frac{\beta \operatorname{sgn}(x^{(i)} - b^{(i)})}{\Gamma(1-\beta)} \int_{b^{(i)}}^{x^{(i)}} \frac{\phi^{(i)}_x(t)}{|x^{(i)} - t^{(i)}|^{1+\beta}} \, dt^{(i)} \\
&\le \frac{\lambda_{\max}(\mathbf{L})|x^{(i)} - b^{(i)}|^{1-\beta}}{2\Gamma(1-\beta)} + \frac{\beta \, \lambda_{\max}(\mathbf{L})\operatorname{sgn}(x^{(i)} - b^{(i)})}{2\Gamma(1-\beta)} \int_{b^{(i)}}^{x^{(i)}} |x^{(i)} - t^{(i)}|^{1-\beta} \, dt^{(i)} \\
&= \frac{\lambda_{\max}(\mathbf{L})|x^{(i)} - b^{(i)}|^{2-\beta}}{2\Gamma(1-\beta)} + \frac{\beta \, \lambda_{\max}(\mathbf{L})\operatorname{sgn}(x^{(i)} - b^{(i)})}{2\Gamma(1-\beta)} \cdot \frac{|x^{(i)} - b^{(i)}|^{2-\beta}}{2-\beta} \\
&= \frac{\lambda_{\max}(\mathbf{L})}{2\Gamma(1-\beta)}|x^{(i)} - b^{(i)}|^{2-\beta} \left(1 + \frac{\beta}{2-\beta}\right) \\
&= \frac{\lambda_{\max}(\mathbf{L})}{\Gamma(1-\beta)(2-\beta)}|x^{(i)} - b^{(i)}|^{2-\beta}.
\end{aligned}$$

The other direction of the inequality follows by the same logic, using instead $\phi^{(i)}_x(t) \ge \frac{-\lambda_{\max}(\mathbf{L})}{2}|x^{(i)} - t^{(i)}|^2$ and applying $\ge$ instead of $\le$. □

Proposition 1. *If $f : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable and $\boldsymbol{L}$ smooth, $\beta \in (0, 1]$, then*

$$|\nabla^{(i)} f(x) - \partial^{\beta,\delta}_{b^{(i)}} f(x)| \le K|x^{(i)} - b^{(i)}| \quad \forall i \in [d] \tag{26}$$

where $K = \frac{\lambda_{max}(\boldsymbol{L})(1-\beta)}{(2-\beta)}$

Proof. Using Lemma 4, rearranging terms gives this bound directly. □

Proof of Theorem 1

Theorem 1. *Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable, $\boldsymbol{L}$-smooth, and satisfies Assumptions 1 – 3. Let $\beta \in (0, 1)$. Define K as in Lemma 1. Then, for each $t \ge 0$*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(x_t)\|^2_{\boldsymbol{D}}\right] \le \frac{(f(x_0) - f^*)}{cT} \tag{27}$$

if one of the below conditions is true:

- *The vectors x_t and b_t are the iterates of CFGD-1 and (9) respectively, $\mu \in \left(\frac{-4.236}{K}, \frac{0.236}{K}\right)$ and $\boldsymbol{D}$ satisfies (11);*
- *The vectors x_t and b_t are the iterates of CFGD-2 and (9) respectively, $\mu \in \left(\frac{-4.236}{K}, \frac{0.236}{K}\right)$ and $\boldsymbol{D}$ satisfies (12).*

where $c = \left\{(1 - K\mu) - \frac{(1+K\mu)^2}{2}\right\}$

Proof. Throughout the proof we will use the notation $[v_i]$ to denote the vector of d elements with i^{th} element v_i. We note that all the results for single variable f hold since f also satisfies the single variable $\lambda_{max}(\mathbf{L})$-smooth definition in each component (Lemma 1 and Lemma 2). We start with $\mathbf{L}$-smooth property of f.

(i) From Assumption 2 with $x = x_{t+1} = x_t - \mathbf{D}\mathbf{A}_t \nabla f(x_t)$ and $y = x_t$, we get

$$\begin{aligned}
\mathbb{E}\left[f(x_{t+1}) \mid x_t\right] \le \quad & \mathbb{E}\left[f(x_t) + \left\langle \nabla f(x_t), -\mathbf{D}\mathbf{A}_t \partial^{\beta,\delta}_{b_t} f(x_t)\right\rangle\right] + \frac{1}{2}\mathbb{E}\left[\left\langle \mathbf{L}\left(-\mathbf{D}\mathbf{A}_t \partial^{\beta,\delta}_{b_t} f(x_t)\right), -\mathbf{D}\mathbf{A}_t \partial^{\beta,\delta}_{b_t} f(x_t)\right\rangle \mid x_t\right] \\
= \quad & f(x_t) - \left\langle \nabla f(x_t), \mathbf{D}\mathbb{E}[\mathbf{A}_t] \partial^{\beta,\delta}_{b_t} f(x_t)\right\rangle + \frac{1}{2}\left\langle \mathbb{E}\left[\mathbf{A}_t \mathbf{D}\mathbf{L}\mathbf{D}\mathbf{A}_t\right] \partial^{\beta,\delta}_{b_t} f(x_t), \partial^{\beta,\delta}_{b_t} f(x_t)\right\rangle.
\end{aligned}$$

From the unbiasedness of the sketch $\mathbf{A}_t$,

$$\begin{aligned}
\mathbb{E}\left[f(x_{t+1}) \mid x_t\right] \le\ & f(x_t) - \left\langle \nabla f(x_t), \mathbf{D}\partial_{b_t}^{\beta,\delta} f(x_t)\right\rangle + \frac{1}{2}\mathbb{E}\left[\langle \mathbf{A}_t\mathbf{DLDA}_t\rangle\, \partial_{b_t}^{\beta,\delta} f(x_t), \partial_{b_t}^{\beta,\delta} f(x_t)\right] \\
\overset{(11)}{\le}\ & f(x_t) - \left\langle \nabla f(x_t), \mathbf{D}\partial_{b_t}^{\beta,\delta} f(x_t)\right\rangle + \frac{1}{2}\left\langle \mathbf{D}\partial_{b_t}^{\beta,\delta} f(x_t), \partial_{b_t}^{\beta,\delta} f(x_t)\right\rangle \\
\overset{(13)}{\le}\ & f(x_t) - \left\langle \left[\nabla^{(i)} f(x_t)\right], \mathbf{D}\left[\nabla^{(i)} f(x_t) - K\left|x_t^{(i)} - b_t^{(i)}\right|\right]\right\rangle \\
& + \frac{1}{2}\left\langle \mathbf{D}\left[\nabla^{(i)} f(x_t) + K\left|x_t^{(i)} - b_t^{(i)}\right|\right], \left[\nabla^{(i)} f(x_t) + K\left|x_t^{(i)} - b_t^{(i)}\right|\right]\right\rangle \\
\le\ & f(x_t) - \left\langle \left[\nabla^{(i)} f(x_t)\right], \mathbf{D}\left[\nabla^{(i)} f(x_t)\right]\right\rangle + \left\langle \left[\nabla^{(i)} f(x_t)\right], \mathbf{D}K\left|x_t^{(i)} - b_t^{(i)}\right|\right\rangle \\
& + \frac{1}{2}\left\langle \mathbf{D}\left[\nabla^{(i)} f(x_t) + K\left|x_t^{(i)} - b_t^{(i)}\right|\right], \left[\nabla^{(i)} f(x_t) + K\left|x_t^{(i)} - b_t^{(i)}\right|\right]\right\rangle \\
\overset{(9)}{\le}\ & f(x_t) - \langle \nabla f(x_t), \mathbf{D}\nabla f(x_t)\rangle + \langle \nabla f(x_t), \mathbf{D}K\mu\nabla f(x_t)\rangle \\
& + \frac{1}{2}\left\langle \mathbf{D}\left[\nabla^{(i)} f(x_t) + K\mu\nabla^{(i)} f(x_t)\right], \left[\nabla^{(i)} f(x_t) + K\mu\nabla^{(i)} f(x_t)\right]\right\rangle \\
\le\ & f(x_t) - \langle \nabla f(x_t), \mathbf{D}\nabla f(x_t)\rangle + \langle \nabla f(x_t), \mathbf{D}K\mu\nabla f(x_t)\rangle \\
& + \frac{1}{2}\langle \mathbf{D}(1+K\mu)\nabla f(x_t), (1+K\mu)\nabla f(x_t)\rangle \\
\le\ & f(x_t) - (1-K\mu)\langle \nabla f(x_t), \mathbf{D}\nabla f(x_t)\rangle + \frac{1}{2}(1+K\mu)^2\langle \nabla f(x_t), \mathbf{D}\nabla f(x_t)\rangle \\
\le\ & f(x_t) - \left\{(1-K\mu) - \frac{(1+K\mu)^2}{2}\right\}\langle \nabla f(x_t), \mathbf{D}\nabla f(x_t)\rangle \\
\le\ & f(x_t) - \left\{(1-K\mu) - \frac{(1+K\mu)^2}{2}\right\}\|f(x_t)\|_{\mathbf{D}}^2.
\end{aligned} \tag{28}$$

We can observe that the term $\left((1-K\mu) - \frac{(1+K\mu)^2}{2}\right)$ is a quadratic expression in μ. To prove convergence, we require this term to be positive. Ensuring the positivity of this entire expression by adjusting μ is a standard technique, similar to the commonly used approaches for proving the convergence of SGD, such as imposing step-size conditions like $\gamma_t \le 1/L$. This practice is well-established and aligns with typical strategies employed in optimisation theory. By solving the quadratic inequality, one can determine that μ should lie in the interval $\mu \in \left(\frac{-4.236}{K}, \frac{0.236}{K}\right)$ for the term to be positive. We denote the term $\left((1-K\mu) - \frac{(1+K\mu)^2}{2}\right)$ by c.

Next, by substracting f^* from both side of (28), taking expectation and applying the tower property, we get

$$\begin{aligned}
\mathbb{E}\left[f(x_{t+1})\right] - f^* =\ & \mathbb{E}\left[\mathbb{E}\left[f(x_{t+1}) \mid x_t\right]\right] - f^* \\
\overset{(28)}{\le}\ & \mathbb{E}\left[f(x_t) - c\|f(x_t)\|_{\mathbf{D}}^2\right] - f^* \\
=\ & \mathbb{E}[f(x_t)] - f^* - c\mathbb{E}\left[\|f(x_t)\|_{\mathbf{D}}^2\right].
\end{aligned}$$

We let $\Delta_{t+1} := \mathbb{E}\left[f(x_{t+1})\right] - f^*$, the last inequality can be written as $\Delta_{t+1} \le \Delta_t - c\mathbb{E}\left[\|f(x_t)\|_{\mathbf{D}}^2\right]$. Summing these inequalities for $t = 0, 1, \ldots, T-1$, we get a telescoping effect leading to

$$\Delta_T \le \Delta_0 - c\sum_{t=0}^{T-1}\mathbb{E}\left[\|f(x_t)\|_{\mathbf{D}}^2\right].$$

To rearrange the terms of this inequality, divide both sides by $T|\mathbf{D}|^{1/d}$, and use the inequality $\Delta_T \ge 0$.

(ii) Similar to the previous case, from Assumption 2 with $x = x_{t+1} = x_t - \mathbf{B}_t\mathbf{D}\nabla f(x_t)$ and $y = x_t$, we get

$$\begin{aligned}
\mathbb{E}\left[f(x_{t+1}) \mid x_t\right] \le\ & \mathbb{E}\left[f(x_t) + \left\langle \nabla f(x_t), -\mathbf{B}_t\mathbf{D}\partial_{b_t}^{\beta,\delta} f(x_t)\right\rangle\right] + \frac{1}{2}\mathbb{E}\left[\left\langle \mathbf{L}\left(-\mathbf{B}_t\mathbf{D}\partial_{b_t}^{\beta,\delta} f(x_t)\right), -\mathbf{B}_t\mathbf{D}\partial_{b_t}^{\beta,\delta} f(x_t)\right\rangle \mid x_t\right] \\
=\ & f(x_t) - \left\langle \nabla f(x_t), \mathbb{E}[\mathbf{B}_t]\mathbf{D}\partial_{b_t}^{\beta,\delta} f(x_t)\right\rangle + \frac{1}{2}\left\langle \mathbb{E}\left[\mathbf{D}(\mathbf{B}_t)^\top\mathbf{L}\mathbf{B}_t\mathbf{D}\right]\partial_{b_t}^{\beta,\delta} f(x_t), \partial_{b_t}^{\beta,\delta} f(x_t)\right\rangle.
\end{aligned}$$

From Assumption 2 and (12), we have

$$\mathbb{E}\left[f(x_{t+1}) \mid x_t\right] \le\ f(x_t) - \left\langle \nabla f(x_t), \mathbf{D}\partial_{b_t}^{\beta,\delta} f(x_t)\right\rangle + \frac{1}{2}\left\langle \mathbf{D}\partial_{b_t}^{\beta,\delta} f(x_t), \partial_{b_t}^{\beta,\delta} f(x_t)\right\rangle$$

The remainder of the proof follows a similar procedure as previously demonstrated in **(i)**. Thus, we obtain the same upper bound on $\mathbb{E}\left[f(x_{t+1}) \mid x_t\right]$ as in (28). Following the steps from the first part, we conclude the proof. □

Proof of Proposition 2

Before beginning to prove the proposition, we introduce a useful lemma about positive definite matrices.

Lemma 5. *For positive definite matrices $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, and $\boldsymbol{L}$, the following inequality holds:*

$$\boldsymbol{D}_1\boldsymbol{L}\boldsymbol{D}_2 + \boldsymbol{D}_2\boldsymbol{L}\boldsymbol{D}_1 \leq \boldsymbol{D}_1\boldsymbol{L}\boldsymbol{D}_1 + \boldsymbol{D}_2\boldsymbol{L}\boldsymbol{D}_2.$$

Proof. Given $\mathbf{D}_1$, $\mathbf{D}_2$, and $\mathbf{L}$ are positive definite matrices, for any nonzero vector x, we have $x^\top \mathbf{D}_1 x > 0$, $x^\top \mathbf{D}_2 x > 0$, and $x^\top \mathbf{L} x > 0$. Let's analyse the matrix $\mathbf{M}$ given by:

$$\mathbf{M} = \mathbf{D}_1\mathbf{L}\mathbf{D}_1 + \mathbf{D}_2\mathbf{L}\mathbf{D}_2 - \mathbf{D}_1\mathbf{L}\mathbf{D}_2 - \mathbf{D}_2\mathbf{L}\mathbf{D}_1.$$

We need to show that $\mathbf{M}$ is positive semidefinite. This means we need to show that for any nonzero vector x, $x^\top \mathbf{M} x \geq 0$.
Consider the quadratic form:

$$x^\top \mathbf{M} x = x^\top(\mathbf{D}_1\mathbf{L}\mathbf{D}_1 + \mathbf{D}_2\mathbf{L}\mathbf{D}_2 - \mathbf{D}_1\mathbf{L}\mathbf{D}_2 - \mathbf{D}_2\mathbf{L}\mathbf{D}_1)x.$$

Let's rewrite the terms in the quadratic form using the substitution $y = \mathbf{D}_1 x$ and $z = \mathbf{D}_2 x$:

$$x^\top \mathbf{D}_1\mathbf{L}\mathbf{D}_1 x = y^\top \mathbf{L} y,$$

$$x^\top \mathbf{D}_2\mathbf{L}\mathbf{D}_2 x = z^\top \mathbf{L} z,$$

$$x^\top \mathbf{D}_1\mathbf{L}\mathbf{D}_2 x = y^\top \mathbf{L}\mathbf{D}_2 x = y^\top \mathbf{L} z,$$

$$x^\top \mathbf{D}_2\mathbf{L}\mathbf{D}_1 x = z^\top \mathbf{L}\mathbf{D}_1 x = z^\top \mathbf{L} y.$$

Using these substitutions, the quadratic form becomes:

$$x^\top \mathbf{M} x = y^\top \mathbf{L} y + z^\top \mathbf{L} z - y^\top \mathbf{L} z - z^\top \mathbf{L} y.$$

Since $\mathbf{L}$ is positive definite, $\mathbf{L}$ can be written as $\mathbf{L} = \mathbf{U}^\top \mathbf{U}$ for some invertible matrix $\mathbf{U}$. Substituting $\mathbf{L} = \mathbf{U}^\top \mathbf{U}$ into the quadratic form, we get:

$$y^\top \mathbf{L} y = y^\top \mathbf{U}^\top \mathbf{U} y = (\mathbf{U}y)^\top(\mathbf{U}y) = \|\mathbf{U}y\|^2,$$

$$z^\top \mathbf{L} z = z^\top \mathbf{U}^\top \mathbf{U} z = (\mathbf{U}z)^\top(\mathbf{U}z) = \|\mathbf{U}z\|^2,$$

$$y^\top \mathbf{L} z = y^\top \mathbf{U}^\top \mathbf{U} z = (\mathbf{U}y)^\top(\mathbf{U}z).$$

Thus, the quadratic form becomes:

$$x^\top \mathbf{M} x = \|\mathbf{U}y\|^2 + \|\mathbf{U}z\|^2 - (\mathbf{U}y)^\top(\mathbf{U}z) - (\mathbf{U}z)^\top(\mathbf{U}y).$$

Since $(\mathbf{U}y)^\top(\mathbf{U}z)$ is a scalar, we have:

$$(\mathbf{U}y)^\top(\mathbf{U}z) = (\mathbf{U}z)^\top(\mathbf{U}y).$$

Therefore, the quadratic form simplifies to:

$$x^\top \mathbf{M} x = \|\mathbf{U}y\|^2 + \|\mathbf{U}z\|^2 - 2(\mathbf{U}y)^\top(\mathbf{U}z) = \|\mathbf{U}y - \mathbf{U}z\|^2.$$

Since $\|\mathbf{U}y - \mathbf{U}z\|^2 \geq 0$ for all vectors y and z, we have:

$$x^\top \mathbf{M} x = \|\mathbf{U}y - \mathbf{U}z\|^2 \geq 0.$$

Thus, $x^\top \mathbf{M} x \geq 0$ for any nonzero vector x, which means that $\mathbf{M}$ is positive semidefinite. Hence, we have:

$$\mathbf{D}_1\mathbf{L}\mathbf{D}_2 + \mathbf{D}_2\mathbf{L}\mathbf{D}_1 \leq \mathbf{D}_1\mathbf{L}\mathbf{D}_1 + \mathbf{D}_2\mathbf{L}\mathbf{D}_2.$$

This completes the proof. □

Proposition 2. *The optimisation problem (16) w.r.t stepsize matrix $\boldsymbol{D} \in \mathcal{S}^d_{++}$, is a concave optimisation problem with convex set.*

Proof. Given, $\mathbf{L} \in \mathcal{S}^d_{++}$, $\mathbf{A}_t \overset{i.i.d}{\sim} \mathcal{A}$ and $\mathbb{E}[\mathbf{A}_t] = \mathbf{I}_d \; \forall t$, we have to prove that the the set $\mathcal{X} := \left\{\mathbf{D} \mid \mathbb{E}\left[\mathbf{A}_t\mathbf{DLDA}_t\right] \preceq \mathbf{D}, \mathbf{D} \in \mathcal{S}^d_{++}\right\}$ is convex.

Let $(\mathbf{D}_1, \mathbf{D}_2) \in \mathcal{X}$. Which implies:

$$\mathbb{E}\left[\mathbf{A}_t\mathbf{D}_1\mathbf{L}\mathbf{D}_1\mathbf{A}_t\right] \preceq \quad \mathbf{D}_1 \tag{29}$$

$$\mathbb{E}\left[\mathbf{A}_t\mathbf{D}_2\mathbf{L}\mathbf{D}_2\mathbf{A}_t\right] \preceq \quad \mathbf{D}_2 \tag{30}$$

We have to show that $\alpha\mathbf{D}_1 + (1-\alpha)\mathbf{D}_2 \in \mathcal{X}$ for all $\alpha \in (0,1)$. For that, the matrix $\alpha\mathbf{D}_1 + (1-\alpha)\mathbf{D}_2$ need to satisfy the two properties of set $\mathcal{X}$. **(i)** $\mathbb{E}[\mathbf{A}_t(\alpha\mathbf{D}_1 + (1-\alpha)\mathbf{D}_2)\mathbf{L}(\alpha\mathbf{D}_1 + (1-\alpha)\mathbf{D}_2)\mathbf{A}_t] \preceq \alpha\mathbf{D}_1 + (1-\alpha)\mathbf{D}_2$ and **(ii)** $\alpha\mathbf{D}_1 + (1-\alpha)\mathbf{D}_2$ is a positive definite matrix. We start with **(i)** below.

$$\begin{aligned}
\mathbb{E}[\mathbf{A}_t(\alpha\mathbf{D}_1 + (1-\alpha)\mathbf{D}_2)\mathbf{L}(\alpha\mathbf{D}_1 + (1-\alpha)\mathbf{D}_2)\mathbf{A}_t] = \;& \mathbb{E}[\alpha^2\mathbf{A}_t\mathbf{D}_1\mathbf{L}\mathbf{D}_1\mathbf{A}_t + (1-\alpha)^2\mathbf{A}_t\mathbf{D}_2\mathbf{L}\mathbf{D}_2\mathbf{A}_t \\
& + \alpha(1-\alpha)\mathbf{A}_t\mathbf{D}_1\mathbf{L}\mathbf{D}_2\mathbf{A}_t + \alpha(1-\alpha)\mathbf{A}_t\mathbf{D}_2\mathbf{L}\mathbf{D}_1\mathbf{A}_t] \\
\preceq \;& \mathbb{E}[\alpha^2\mathbf{A}_t\mathbf{D}_1\mathbf{L}\mathbf{D}_1\mathbf{A}_t + (1-\alpha)^2\mathbf{A}_t\mathbf{D}_2\mathbf{L}\mathbf{D}_2\mathbf{A}_t \\
& + \alpha(1-\alpha)\mathbf{A}_t(\mathbf{D}_1\mathbf{L}\mathbf{D}_1 + \mathbf{D}_2\mathbf{L}\mathbf{D}_2)\mathbf{A}_t] \quad \text{(Using Lemma 5)} \\
\preceq \;& \alpha^2\mathbb{E}[\mathbf{A}_t\mathbf{D}_1\mathbf{L}\mathbf{D}_1\mathbf{A}_t] + (1-\alpha)^2\mathbb{E}[\mathbf{A}_t\mathbf{D}_2\mathbf{L}\mathbf{D}_2\mathbf{A}_t] \\
& + \alpha(1-\alpha)\mathbb{E}[\mathbf{A}_t(\mathbf{D}_1\mathbf{L}\mathbf{D}_1 + \mathbf{D}_2\mathbf{L}\mathbf{D}_2)\mathbf{A}_t] \quad \text{(Using linearity of expectation)} \\
\preceq \;& \alpha^2\mathbb{E}[\mathbf{A}_t\mathbf{D}_1\mathbf{L}\mathbf{D}_1\mathbf{A}_t] + (1-\alpha)^2\mathbb{E}[\mathbf{A}_t\mathbf{D}_2\mathbf{L}\mathbf{D}_2\mathbf{A}_t] \\
& + \alpha(1-\alpha)\mathbb{E}[\mathbf{A}_t\mathbf{D}_1\mathbf{L}\mathbf{D}_1\mathbf{A}_t] + \alpha(1-\alpha)\mathbb{E}[\mathbf{A}_t\mathbf{D}_2\mathbf{L}\mathbf{D}_2\mathbf{A}_t] \\
\preceq \;& (\alpha^2 + \alpha(1-\alpha)\mathbb{E}[\mathbf{A}_t\mathbf{D}_1\mathbf{L}\mathbf{D}_1\mathbf{A}_t] + ((1-\alpha)^2 + \alpha(1-\alpha))\mathbb{E}[\mathbf{A}_t\mathbf{D}_2\mathbf{L}\mathbf{D}_2\mathbf{A}_t] \\
\preceq \;& (\alpha^2 + \alpha(1-\alpha)\mathbf{D}_1 + ((1-\alpha)^2 + \alpha(1-\alpha))\mathbf{D}_2 \quad \text{(Using (29) and (30))} \\
\preceq \;& \alpha\mathbf{D}_1 + (1-\alpha)\mathbf{D}_2
\end{aligned}$$

For **(ii)**, proving that the matrix $\alpha\mathbf{D}_1 + (1-\alpha)\mathbf{D}_2$ is positive definite is trivial. Hence, we conclude our proof. □

B Layer-wise case

In this section, we examine the block-diagonal structure of $\mathbf{L}$ for both CFGD-1 and CFGD-2. Specifically, we introduce hyperparameters for CFGD-1 that are optimised for neural network training. Assume $\mathbf{L}$ is block-diagonal, represented by $\mathbf{L} = \mathrm{Diag}(\mathbf{L}_1, \ldots, \mathbf{L}_l)$, where each $\mathbf{L}_i$ is a positive definite matrix. This setup generalises the traditional smoothness condition; for example, in simpler cases, each $\mathbf{L}_i$ could be a scaled identity matrix $L\mathbf{I}_{d_i}$. Inspired by (Li, Karagulyan, and Richtárik 2023a), we also use block-diagonal structures for the sketches and the stepsize matrices, setting $\mathbf{D} = \mathrm{Diag}(\mathbf{D}_1, \ldots, \mathbf{D}_l)$ and $\mathbf{A}_t = \mathrm{Diag}(\mathbf{A}_{t1}, \ldots, \mathbf{A}_{tk})$, with each $\mathbf{D}_i$ and $\mathbf{A}_{ik}$ being positive definite. Note that the left side of the inequality in (16) is quadratic in $\mathbf{D}$, while the right side depends linearly on $\mathbf{D}$. Therefore, for any positive definite matrix $\mathbf{W}$, there exists a scalar $\omega > 0$ such that:

$$\omega^2\lambda_{\max}\left(\mathbb{E}\left[\mathbf{A}_t\mathbf{WLWA}_t\right]\right) \leq \omega\lambda_{\min}(\mathbf{W}).$$

Thus, for the scaled matrix $\omega\mathbf{W}$, we deduce:

$$\begin{aligned}
\mathbb{E}\left[\mathbf{A}_t(\omega\mathbf{W})\mathbf{L}(\omega\mathbf{W})\mathbf{A}_t\right] &\preceq \omega^2\lambda_{\max}\left(\mathbb{E}\left[\mathbf{A}_t\mathbf{WLWA}_t\right]\right)\mathbf{I}_d \\
&\preceq \omega\lambda_{\min}(\mathbf{W})\mathbf{I}_d \preceq \omega\mathbf{W}.
\end{aligned} \tag{31}$$

The following theorem is based on applying this observation to the blocks of matrices $\mathbf{D}$, $\mathbf{L}$, and $\mathbf{A}_t$ within CFGD-1.

Proof of Theorem 2

Theorem 2. *Suppose $f : \mathbb{R}^d \to \mathbb{R}$ meets Assumptions 1 and 2, and $\boldsymbol{L}$ has a layer-separable structure such that $\boldsymbol{L} = \mathrm{Diag}(\boldsymbol{L}_1, \ldots, \boldsymbol{L}_l)$, with each $\boldsymbol{L}_i \in \mathcal{S}^{d_i}_{++}$. Choose random matrices $\boldsymbol{A}_{t1}, \ldots, \boldsymbol{A}_{tl} \in \mathcal{S}^d_+$ that satisfy Assumption 3 for each layer $i = [l]$, and set $\boldsymbol{A}_t = \mathrm{Diag}(\boldsymbol{A}_{t1}, \ldots, \boldsymbol{A}_{tl})$. Additionally, select matrices $\boldsymbol{W}_1, \ldots, \boldsymbol{W}_l \in \mathcal{S}^d_{++}$ and positive scalars $\omega_1, \ldots, \omega_l$ such that:*

$$\omega_i \leq \lambda^{-1}_{max}\left(\mathbb{E}\left[\boldsymbol{W}_i^{-1/2}\boldsymbol{A}_{ti}\boldsymbol{W}_i\boldsymbol{L}_i\boldsymbol{W}_i\boldsymbol{A}_{ti}\boldsymbol{W}_i^{-1/2}\right]\right) \quad \forall i \in [l]. \tag{32}$$

Define $\boldsymbol{W} := \mathrm{Diag}(\boldsymbol{W}_1, \ldots, \boldsymbol{W}_l)$, $\Omega := \mathrm{Diag}(\omega_1\boldsymbol{I}_{d_1}, \ldots, \omega_l\boldsymbol{I}_{d_l})$, and $\boldsymbol{D} = \Omega W$. Then, for some $c > 0$ we have:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(x_t)\|^2_{\frac{\Omega\boldsymbol{W}}{|\Omega\boldsymbol{W}|^{1/d}}}\right] \leq \frac{(f(x_0) - f^*)}{c|\Omega\boldsymbol{W}|^{1/d}T}. \tag{33}$$

Proof. Note that

$$\mathbb{E}\left[\mathbf{A}_t\mathbf{DLDA}_t\right] = \mathrm{Diag}(\mathbf{Q}_{t1}, \ldots, \mathbf{Q}_{tl}),$$

where

$$\mathbf{Q}_{ti} = \omega_i^2 \mathbb{E}\left[\mathbf{A}_{ti}\mathbf{W}_i\mathbf{L}_i\mathbf{D}_i\mathbf{A}_{ti}\right].$$

In other words,

$$\mathbb{E}\left[\mathbf{A}_t\mathbf{DLDA}_t\right] = \begin{pmatrix} \mathbf{Q}_{t1} & 0 & \cdots & 0 \\ 0 & \mathbf{Q}_{t2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{Q}_{tl} \end{pmatrix},$$

which means that (11) holds iff $\mathbf{Q}_{ti} \preceq \omega_i \mathbf{W}_i$ for all $i \in [l]$, which holds iff (32) holds. Therefore, Theorem 1 applies, and we conclude that

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(x_t)\|^2_{\Omega\mathbf{W}}\right] \leq \frac{(f(x_0) - f^*)}{cT}. \tag{34}$$

Here, $c > 0$ is from Theorem 1. To obtain (33), it remains to multiply both sides of (34) by $\frac{1}{|\Omega\mathbf{W}|^{1/d}}$. □

In particular, if the scalars $\{\omega_i\}_{i=1}^l$ are chosen to be equal to their maximum allowed values from (32), then the convergence factor of (33) is equal to

$$|\Omega\mathbf{W}|^{\frac{-1}{d}} = \left[\prod_{i=1}^{l} \lambda_{max}^{d_i}\left(\mathbb{E}\left[\mathbf{W}_i^{-1/2}\mathbf{A}_{ti}\mathbf{W}_i\mathbf{L}_i\mathbf{W}_i\mathbf{A}_{ti}\mathbf{W}_i^{-1/2}\right]\right)\right] |\mathbf{W}|^{\frac{-1}{d}}.$$

The setup of Theorem 2 is adapted from (Li, Karagulyan, and Richtárik 2023a).

Interpretations from Table 1

Table 1: Summary of communication complexities of CFGD-1 and CFGD-2 with different sketches and stepsize matrices. The $\mathbf{D}_i$ here for CFGD-1 is $\mathbf{W}_i$ with the optimal scaling determined using Theorem 2, for CFGD-2 it is the optimal stepsize matrix defined in (18). The constant $\frac{(f(x_0)-f^*)}{c\epsilon^2}$ is hidden, l is the number of layers, k_i is the mini-batch size for the i-th layer if we use the rand-k sketch. The notation $\tilde{\mathbf{L}}_{i,k}$ is defined as $\frac{d-k}{d-1}\operatorname{diag}(\mathbf{L}_i) + \frac{k-1}{d-1}\mathbf{L}_i$. This table is taken from (Li, Karagulyan, and Richtárik 2023a), but it perfectly fits in our case and the proposed algorithms.

No.	**The method**	$(\mathbf{A}_{it}, \mathbf{D}_i)$	$l \geq 1, d_i, k_i, \sum_{i=1}^{l} k_i = k$, layer structure	$l = 1, k$, general structure
1	CFGD-1	$\left(\mathbf{I}_d, \omega\mathbf{L}_i^{-1}\right)$	$d \cdot \lvert\mathbf{L}\rvert^{1/d}$	$d \cdot \lvert\mathbf{L}\rvert^{1/d}$
2	CFGD-1	$\left(\mathbf{I}_d, \omega \operatorname{diag}^{-1}(\mathbf{L}_i)\right)$	$d \cdot \lvert\operatorname{diag}(\mathbf{L})\rvert^{1/d}$	$d \cdot \lvert\mathbf{L}\rvert^{1/d}$
3	CFGD-1	$(\mathbf{I}_d, \omega\mathbf{I}_{d_i})$	$d \cdot \left(\prod_{i=1}^{l} \lambda_{max}^{d_i}(\mathbf{L}_i)\right)^{1/d}$	$d \cdot \lambda_{\max}(\mathbf{L})$
4	CFGD-1	$(\textbf{rand-}1, \omega\mathbf{I}_{d_i})$	$l \cdot \left(\prod_{i=1}^{l} d_i^{d_i} \left(\max_j(\mathbf{L}_i)_{jj}\right)^{d_i}\right)^{1/d}$	$d \cdot \max_j(\mathbf{L}_{jj})$
5	CFGD-1	$\left(\textbf{rand-}1, \omega\mathbf{L}_i^{-1}\right)$	$l \cdot \left(\frac{\prod_{i=1}^{l} d_i^{d_i} \lambda_{max}^{d_i}\left(\mathbf{L}_i^{\frac{1}{2}} \operatorname{diag}\left(\mathbf{L}_i^{-1}\right)\mathbf{L}_i^{\frac{1}{2}}\right)}{\prod_{i=1}^{l} \lvert\mathbf{L}^{-1}\rvert}\right)$	$d \cdot \frac{\lambda_{max}\left(\mathbf{L}^{\frac{1}{2}} \operatorname{diag}\left(\mathbf{L}^{-1}\right)\mathbf{L}^{\frac{1}{2}}\right)}{\lvert\mathbf{L}^{-1}\rvert^{1/d}}$
6	CFGD-1	$\left(\textbf{rand-}1, \omega \operatorname{diag}^{-1}(\mathbf{L}_i)\right)$	$l \cdot \left(\frac{\prod_{i=1}^{l} d_i^{d_i}}{\prod_{j=1}^{d}\left(\mathbf{L}_{jj}^{-1}\right)}\right)$	$d \cdot \lvert\operatorname{diag}(\mathbf{L})\rvert^{1/d}$
7	CFGD-1	$\left(\textbf{rand-}k_i, \omega \operatorname{diag}^{-1}(\mathbf{L}_i)\right)$	$k \cdot \left(\prod_{i=1}^{l} \left(\frac{d_i}{k_i}\right)^{d_i} \lvert\operatorname{diag}(\mathbf{L})\rvert\right)^{1/d}$	$d \cdot \lvert\operatorname{diag}(\mathbf{L})\rvert^{1/d}$
9	CFGD-2	$\left(\mathbf{I}_d, \mathbf{L}_i^{-1}\right)$	$d \cdot \lvert\mathbf{L}\rvert^{1/d}$	$d \cdot \lvert\mathbf{L}\rvert^{1/d}$
10	CFGD-2	$\left(\textbf{rand-}1, \frac{\operatorname{diag}^{-1}(\mathbf{L}_i)}{d_i}\right)$	$l \cdot \left(\prod_{i=1}^{l} d_i^{d_i}\right)^{1/d} \lvert\operatorname{diag}(\mathbf{L})\rvert^{1/d}$	$d \cdot \lvert\operatorname{diag}(\mathbf{L})\rvert^{1/d}$
11	CFGD-2	$\left(\textbf{rand-}k, \frac{k_i}{d_i}\tilde{\mathbf{L}}_{i,k_i}^{-1}\right)$	$k \cdot \left(\prod_{i=1}^{l} \left(\frac{d_i}{k_i}\right)^{\frac{d_i}{d}}\right) \left(\prod_{i=1}^{l} \lvert\tilde{\mathbf{L}}_{i,k_i}\rvert\right)^{1/d}$	$d \cdot \lvert\tilde{\mathbf{L}}_{1,k}\rvert^{1/d}$
12	GD	$\left(\mathbf{I}_d, \lambda_{\max}^{-1}(\mathbf{L})\mathbf{I}_d\right)$	–	$d \cdot \lambda_{\max}(\mathbf{L})$
13	FGD	$\left(\mathbf{I}_d, \lambda_{\max}^{-1}(\mathbf{L})\mathbf{I}_d\right)$	–	$d \cdot \lambda_{\max}(\mathbf{L})$

The communication complexity of the *gradient descent* (GD) algorithm, as shown in row 13, is given by $d\lambda_{\max}(\mathbf{L})$, where $\lambda_{\max}(\mathbf{L})$ represents the smoothness constant of the function. In comparison, CFGD-1 and CFGD-2, which utilize matrix step sizes without compression, given in row 1 and row 9 respectively, exhibit superior performance in both iteration and communication complexities when compared to GD. However, certain results in the table require a more detailed examination, which we provide below. For the rest of this section, we will exclude the constant factor $(f(x_0) - f^*)/c\epsilon^2$ from the communication complexity, as it is common to all scenarios in the table and does not affect comparative analysis.

Comparison of row 5 and 7 We theoretically establish that the communication complexity presented in row 5 is consistently higher than that in row 7. This result is supported by the following proposition.

Proposition 3. *((Li, Karagulyan, and Richtárik 2023a)) For any matrix $\boldsymbol{L} \in \mathcal{S}_{++}^d$, the following inequality holds:*

$$\lambda_{\max}\left(\boldsymbol{L}^{\frac{1}{2}} \mathit{diag}(\boldsymbol{L}^{-1})\boldsymbol{L}^{\frac{1}{2}}\right) \cdot \lvert\boldsymbol{L}\rvert^{\frac{1}{d}} \geq \lvert\operatorname{diag}(\boldsymbol{L})\rvert^{\frac{1}{d}}.$$

Proof. The inequality given above in Proposition 3 can be reformulated as

$$\lambda_{\max}\left(\mathbf{L} \cdot \operatorname{diag}(\mathbf{L}^{-1})\right) \geq \lvert\mathbf{L}^{-1} \cdot \operatorname{diag}(\mathbf{L})\rvert^{\frac{1}{d}}.$$

We use the notation

$$\mathbf{M}_1 = \mathbf{L} \cdot \operatorname{diag}(\mathbf{L}^{-1}), \quad \mathbf{M}_2 = \mathbf{L}^{-1} \cdot \operatorname{diag}(\mathbf{L}),$$

and notice that for any $i \in [d]$, we have

$$(\mathbf{M}_1)_{ii} = (\mathbf{L})_{ii} \cdot (\mathbf{L}^{-1})_{ii} = (\mathbf{M}_2)_{ii}.$$

As a result,

$$\lambda_{\max}(\mathbf{M}_1) \geq \left(\prod_{i=1}^{d}(\mathbf{M}_1)_{ii}\right)^{\frac{1}{d}} = \left(\prod_{i=1}^{d}(\mathbf{M}_2)_{ii}\right)^{\frac{1}{d}} \geq |\mathbf{M}_2|^{\frac{1}{d}},$$

where the first inequality follows from the fact that each diagonal element is upper-bounded by the maximum eigenvalue, while the second one is derived from the fact that the product of the diagonal elements provides an upper bound for the determinant. □

From Proposition 3, it immediately follows that the result in row 7 is better than row 5 in terms of both communication and iteration complexity.

Comparison of row 6 and 7 In this section, we present examples of matrices $\mathbf{L}$ to demonstrate that rows 6 and 7 are not usually comparable. Let $d = 2$ and $\mathbf{L} \in \mathcal{S}^2_{++}$.

If we choose

$$\mathbf{L} = \begin{pmatrix} 16 & 0 \\ 0 & 1 \end{pmatrix},$$

then

$$|\mathrm{diag}(\mathbf{L})|^{\frac{1}{d}} = 4, \quad \lambda_{\max}(\mathbf{L})^{\frac{1}{2}}|\mathbf{L}|^{\frac{1}{2d}} = 8.$$

Again, if we choose

$$\mathbf{L} = \begin{pmatrix} 16 & 3.9 \\ 3.9 & 1 \end{pmatrix},$$

then

$$|\mathrm{diag}(\mathbf{L})|^{\frac{1}{d}} = 4, \quad \lambda_{\max}(\mathbf{L})^{\frac{1}{2}}|\mathbf{L}|^{\frac{1}{2d}} \approx 3.88.$$

From this example, one can conclude that the relationship between rows 6 and 7 can vary depending on the choice of $\mathbf{L}$.

C Distributed Case

Proof of Theorem 2

We first present some simple technical lemmas whose proofs are deferred to Section D of this supplementary material. Let us recall that $\mathbf{D} \in \mathcal{S}^d_{++}$ is the stepsize matrix, $\mathbf{L}, \mathbf{L}_j \in \mathcal{S}^d_{++}$ are the smoothness matrices for f and f_j respectevely, for all $j \in [n]$, where n is the number of clients.

Lemma 6 (Variance Decomposition)**.** *((Li, Karagulyan, and Richtárik 2023a)) For any random vector $x \in \mathbb{R}^d$, and any matrix $\boldsymbol{M} \in \mathcal{S}^d_+$, the following identity holds:*

$$\mathbb{E}\left[\|x - \mathbb{E}[x]\|^2_{\boldsymbol{M}}\right] = \mathbb{E}\left[\|x\|^2_{\boldsymbol{M}}\right] - \|\mathbb{E}[x]\|^2_{\boldsymbol{M}}. \tag{35}$$

Lemma 7. *((Li, Karagulyan, and Richtárik 2023a)) Assume $\{a_j\}_{j=1}^n$ is a set of independent random vectors in $\mathbb{R}^d$, which satisfy*

$$\mathbb{E}[a_j] = 0, \quad \forall j \in [n].$$

Then, for any $\boldsymbol{M} \in \mathbb{S}^d_+$, we have

$$\mathbb{E}\left[\left\|\frac{1}{n}\sum_{j=1}^{n} a_j\right\|^2_{\boldsymbol{M}}\right] = \frac{1}{n^2}\sum_{j=1}^{n}\mathbb{E}\left[\|a_j\|^2_{\boldsymbol{M}}\right]. \tag{36}$$

Lemma 8. *((Li, Karagulyan, and Richtárik 2023a)) For any vector $x \in \mathbb{R}^d$, and sketch matrix $\boldsymbol{S} \in \mathcal{S}^d_+$ taken from some distribution $\boldsymbol{S}$ over $\mathcal{S}^d_+$, which satisfies*

$$\mathbb{E}[\boldsymbol{S}] = \boldsymbol{I}_d,$$

then for any matrix $\boldsymbol{M} \in \mathcal{S}^d_+$, we have the following identity:

$$\mathbb{E}\left[\|\boldsymbol{S}x - x\|^2_{\boldsymbol{M}}\right] = \|x\|^2_{\mathbb{E}[\boldsymbol{SMS}]-\boldsymbol{M}}. \tag{37}$$

Lemma 9. *((Li, Karagulyan, and Richtárik 2023a)) If we have a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, that is $\boldsymbol{L}$-matrix smooth and lower bounded by f^*, if we assume $\boldsymbol{L} \in \mathcal{S}^d_+$, then the following inequality holds:*

$$\langle \nabla f(x), \boldsymbol{L}^{-1}\nabla f(x)\rangle \leq 2\big(f(x) - f^*\big). \tag{38}$$

Theorem 3. *Let $f_j : \mathbb{R}^d \to \mathbb{R}$ satisfy Assumption 4 and let f satisfy Assumption 1 and 2 with smoothness matrix $\mathbf{L}$. Let $\beta \in (0,1)$. Define $K_j \quad \forall j \in [n]$ as in Lemma 1. If the following conditions are satisfied*

- ***DLD*** $\preceq$ ***D***
- $|x_t^{(i,j)} - b_t^{(i,j)}| = \mu_j \left|\frac{\partial f}{\partial x^{(i,j)}}(x_t)\right|, \quad \mu_j \in \left(\frac{-4.236}{K_j}, \frac{0.236}{K_j}\right), \quad \forall i \in [d], j \in [n].$

then, for some $c, a > 0$, the following convergence bound is true for the iterates of CFGD-1:

$$\min_{0 \le t \le T-1} \mathbb{E}\left[\|\nabla f(x_t)\|^2_{\frac{\boldsymbol{D}}{|\boldsymbol{D}|^{1/d}}}\right] \le \frac{\left(1 + \frac{a^2 \lambda_{\boldsymbol{D}}}{n}\right)^T (f(x_0) - f^*)}{c|\boldsymbol{D}|^{1/d} T} + \frac{a^2 \lambda_{\boldsymbol{D}} \Delta^*}{c|\boldsymbol{D}|^{1/d} n}.$$

where $\Delta^* := f^* - \frac{1}{n}\sum_{j=1}^n f_j^*$,

$$\lambda_{\boldsymbol{D}} := \underset{j}{max}\left\{\lambda_{max}\left(\mathbb{E}\left[\boldsymbol{L}_j^{\frac{1}{2}}\left(\boldsymbol{A}_{tj} - \boldsymbol{I}_d\right)\boldsymbol{DLD}\left(\boldsymbol{A}_{tj} - \boldsymbol{I}_d\right)\boldsymbol{L}_j^{\frac{1}{2}}\right]\right)\right\}$$

and $a^2 := \underset{j}{max}(1 + K_j \mu_j)^2$.

Proof. Let the gradient estimator of our algorithm be defined as:

$$g(x_t) := \frac{1}{n}\sum_{j=1}^n \mathbf{A}_{tj} \partial_{b_t}^{\beta,\delta} f_j(x_t), \tag{39}$$

as a result, the update rule in the distributed case can then be written as:

$$x_{t+1} = x_t - \mathbf{D} g(x_t).$$

Notice that we have:

$$\mathbb{E}[g(x_t) \mid x_t] = \frac{1}{n}\sum_{j=1}^n \mathbb{E}[\mathbf{A}_{tj}] \partial_{b_t}^{\beta,\delta} f_j(x_t) = \frac{1}{n}\sum_{j=1}^n \partial_{b_t}^{\beta,\delta} f_j(x_t) = \partial_{b_t}^{\beta,\delta} f(x_t) \tag{40}$$

We start by applying the $\mathbf{L}$-matrix smoothness of f:

$$\begin{aligned} f(x_{t+1}) \le\ & f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t \rangle + \frac{1}{2}\langle \mathbf{L}(x_{t+1} - x_t), (x_{t+1} - x_t)\rangle \\ \le\ & f(x_t) + \langle \nabla f(x_t), -\mathbf{D} g(x_t)\rangle + \frac{1}{2}\langle \mathbf{L}(-\mathbf{D} g(x_t)), (-\mathbf{D} g(x_t))\rangle \\ \le\ & f(x_t) + \langle \nabla f(x_t), -\mathbf{D} g(x_t)\rangle + \frac{1}{2}\langle \mathbf{LD} g(x_t), \mathbf{D} g(x_t)\rangle. \end{aligned}$$

Taking the expectation conditioned on x_t, we get:

$$\begin{aligned} \mathbb{E}[f(x_{t+1}) \mid x_t] \le\ & f(x_t) - \langle \nabla f(x_t), \mathbf{D}\mathbb{E}[g(x_t) \mid x_t]\rangle + \frac{1}{2}\mathbb{E}[\langle \mathbf{LD} g(x_t), \mathbf{D} g(x_t)\rangle \mid x_t] \\ \le\ & f(x_t) - \langle \nabla f(x_t), \mathbf{D}\partial_{b_t}^{\beta,\delta} f(x_t)\rangle + \frac{1}{2}\mathbb{E}\left[\langle \mathbf{LD} g(x_t), \mathbf{D} g(x_t)\rangle \,|\, x_t\right] \\ \overset{(13)}{\le}\ & f(x_t) - \left\langle \left[\nabla^{(i)} f(x_t)\right], \mathbf{D}\left[\nabla^{(i)} f(x_t) - K\left|x_t^{(i)} - b_t^{(i)}\right|\right]\right\rangle + \frac{1}{2}\underbrace{\langle \mathbf{LD} g(x_t), \mathbf{D} g(x_t)\rangle}_{:=P} \end{aligned} \tag{41}$$

Applying Lemma 6 to the term P, we obtain:

$$\begin{aligned} P =\ & \mathbb{E}\left[\|g(x_t)\|^2_{\mathbf{DLD}} \mid x_t\right] \\ =\ & \mathbb{E}\left[\|g(x_t) - \mathbb{E}[g(x_t) \mid x_t]\|^2_{\mathbf{DLD}} \mid x_t\right] + \|\mathbb{E}[g(x_t) \mid x_t]\|^2_{\mathbf{DLD}}. \end{aligned}$$

From the unbiasedness of the sketches, we have $\mathbb{E}[g(x_t) \mid x_t] = \partial_{b_t}^{\beta,\delta} f(x_t)$, which yields:

$$\begin{aligned} P =\ & \mathbb{E}\left[\|g(x_t) - \mathbb{E}[g(x_t) \mid x_t]\|^2_{\mathbf{DLD}} \mid x_t\right] + \|\partial_{b_t}^{\beta,\delta} f(x_t)\|^2_{\mathbf{DLD}} \\ =\ & \mathbb{E}\left[\left\|\frac{1}{n}\sum_{j=1}^n \left(\mathbf{A}_{tj}\partial_{b_t}^{\beta,\delta} f_j(x_t) - \partial_{b_t}^{\beta,\delta} f_j(x_t)\right)\right\|^2_{\mathbf{DLD}} \middle| x_t\right] + \|\partial_{b_t}^{\beta,\delta} f(x_t)\|^2_{\mathbf{DLD}}. \end{aligned}$$

Using Lemma 7, we have:

$$\begin{aligned}
P = \quad & \frac{1}{n^2}\sum_{j=1}^{n}\mathbb{E}\left[\left\|\mathbf{A}_{tj}\partial_{b_t}^{\beta,\delta}f_j(x_t)-\partial_{b_t}^{\beta,\delta}f_j(x_t)\right\|_{\mathbf{DLD}}^2|x_t\right]+\|\partial_{b_t}^{\beta,\delta}f(x_t)\|_{\mathbf{DLD}}^2 \\
\leq \quad & \frac{1}{n^2}\sum_{j=1}^{n}\mathbb{E}\left[\left\|\mathbf{A}_{tj}\partial_{b_t}^{\beta,\delta}f_j(x_t)-\partial_{b_t}^{\beta,\delta}f_j(x_t)\right\|_{\mathbf{DLD}}^2|x_t\right]+\|\partial_{b_t}^{\beta,\delta}f(x_t)\|_{\mathbf{D}}^2, \qquad (42)
\end{aligned}$$

The last inequality holds due to $\mathbf{DLD}\preceq\mathbf{D}$. □

Lemma 10. *((Li, Karagulyan, and Richtárik 2023a)) Let $\boldsymbol{A}\overset{i.i.d}{\sim}\mathcal{A}$ where $\mathcal{A}$ is a distribution over $\mathcal{S}_+^d$ and $\mathbb{E}[\boldsymbol{A}]=\boldsymbol{I}_d$. The following holds for any $x\in\mathbb{R}^d$ and any matrix $\boldsymbol{G}$,*

$$\mathbb{E}\left[\|\boldsymbol{G}x-x\|_{\boldsymbol{DLD}}^2\right]\leq\lambda_{max}\left(\boldsymbol{G}^{1/2}\mathbb{E}\left[(\boldsymbol{G}-\boldsymbol{I}_d)\boldsymbol{DLD}(\boldsymbol{G}-\boldsymbol{I}_d)\right]\boldsymbol{G}^{1/2}\right).\|x\|_{\boldsymbol{G}^{-1}}^2. \qquad (43)$$

Plugging (42) into (41) and applying Lemma 9 and Lemma 10, we can write

$$\begin{aligned}
\mathbb{E}[f(x_{t+1})\mid x_t]\leq \quad & f(x_t)-\left\langle\left[\nabla^{(i)}f(x_t)\right],\mathbf{D}\left[\nabla^{(i)}f(x_t)-K\left|x_t^{(i)}-b_t^{(i)}\right|\right]\right\rangle \\
& +\frac{1}{2n^2}\sum_{j=1}^{n}\mathbb{E}\left[\left\|\mathbf{A}_{tj}\partial_{b_t}^{\beta,\delta}f_j(x_t)-\partial_{b_t}^{\beta,\delta}f_j(x_t)\right\|_{\mathbf{DLD}}^2|x_t\right]+\frac{1}{2}\|\partial_{b_t}^{\beta,\delta}f(x_t)\|_{\mathbf{D}}^2 \\
\overset{(13)}{\leq} \quad & f(x_t)-\left\langle\left[\nabla^{(i)}f(x_t)\right],\mathbf{D}\left[\nabla^{(i)}f(x_t)\right]\right\rangle+\left\langle\left[\nabla^{(i)}f(x_t)\right],\mathbf{D}K\left|x_t^{(i)}-b_t^{(i)}\right|\right\rangle \\
& +\frac{1}{2n^2}\sum_{j=1}^{n}\mathbb{E}\left[\left\|\mathbf{A}_{tj}\partial_{b_t}^{\beta,\delta}f_j(x_t)-\partial_{b_t}^{\beta,\delta}f_j(x_t)\right\|_{\mathbf{DLD}}^2|x_t\right] \\
& +\frac{1}{2}\left\langle\mathbf{D}\left[\nabla^{(i)}f(x_t)+K\left|x_t^{(i)}-b_t^{(i)}\right|\right],\left[\nabla^{(i)}f(x_t)+K\left|x_t^{(i)}-b_t^{(i)}\right|\right]\right\rangle \\
\overset{(9)}{\leq} \quad & f(x_t)-\langle\nabla f(x_t),\mathbf{D}\nabla f(x_t)\rangle+\langle\nabla f(x_t),\mathbf{D}K\mu\nabla f(x_t)\rangle \\
& +\frac{1}{2n^2}\sum_{j=1}^{n}\mathbb{E}\left[\left\|\mathbf{A}_{tj}\partial_{b_t}^{\beta,\delta}f_j(x_t)-\partial_{b_t}^{\beta,\delta}f_j(x_t)\right\|_{\mathbf{DLD}}^2|x_t\right]+\frac{1}{2}\left\langle\mathbf{D}(1+K\mu)\nabla f(x_t),(1+K\mu)\nabla f(x_t)\right\rangle \\
\leq \quad & f(x_t)-(1-K\mu)\|\nabla f(x_t)\|_{\mathbf{D}}^2 \\
& +\frac{1}{2n^2}\sum_{j=1}^{n}\mathbb{E}\left[\left\|\mathbf{A}_{tj}\partial_{b_t}^{\beta,\delta}f_j(x_t)-\partial_{b_t}^{\beta,\delta}f_j(x_t)\right\|_{\mathbf{DLD}}^2|x_t\right]+\frac{(1+K\mu)^2}{2}\|\nabla f(x_t)\|_{\mathbf{D}}^2 \\
\leq \quad & f(x_t)-\underbrace{\left\{(1-K\mu)-\frac{(1+K\mu)^2}{2}\right\}}_{:=c>0\implies\mu\in\left(\frac{-4.236}{K},\frac{0.236}{K}\right)}\|\nabla f(x_t)\|_{\mathbf{D}}^2+\frac{1}{2n^2}\sum_{j=1}^{n}\mathbb{E}\left[\left\|\mathbf{A}_{tj}\partial_{b_t}^{\beta,\delta}f_j(x_t)-\partial_{b_t}^{\beta,\delta}f_j(x_t)\right\|_{\mathbf{DLD}}^2|x_t\right] \\
\overset{(43)}{\leq} \quad & f(x_t)-c\|\nabla f(x_t)\|_{\mathbf{D}}^2+\frac{1}{2n^2}\sum_{j=1}^{n}\lambda_{max}\left(\mathbf{L}_j^{1/2}\mathbb{E}\left[(\mathbf{A}_{tj}-\mathbf{I}_d)\mathbf{DLD}(\mathbf{A}_{tj}-\mathbf{I}_d)\right]\mathbf{L}_j^{1/2}\right).\|\partial_{b_t}^{\beta,\delta}f_j(x_t)\|_{\mathbf{L}_j^{-1}}^2 \\
\overset{(9,13)}{\leq} \quad & f(x_t)-c\|\nabla f(x_t)\|_{\mathbf{D}}^2+\frac{1}{2n^2}\sum_{j=1}^{n}\lambda_{max}\left(\mathbf{L}_j^{1/2}\mathbb{E}\left[(\mathbf{A}_{tj}-\mathbf{I}_d)\mathbf{DLD}(\mathbf{A}_{tj}-\mathbf{I}_d)\right]\mathbf{L}_j^{1/2}\right).(1+K_j\mu_j)^2\|\nabla f_j(x_t)\|_{\mathbf{L}_j^{-1}}^2 \\
\overset{(38)}{\leq} \quad & f(x_t)-c\|\nabla f(x_t)\|_{\mathbf{D}}^2+\frac{1}{n^2}\sum_{j=1}^{n}\lambda_{max}\left(\mathbf{L}_j^{1/2}\mathbb{E}\left[(\mathbf{A}_{tj}-\mathbf{I}_d)\mathbf{DLD}(\mathbf{A}_{tj}-\mathbf{I}_d)\right]\mathbf{L}_j^{1/2}\right)(1+K_j\mu_j)^2(f_j(x_k)-f_j^*(x_k))
\end{aligned}$$

From the definition of $\lambda_{\mathbf{D}}$ and a^2 from Theorem 2, we bound $f(x_{t+1})$ by

$$\begin{aligned}\mathbb{E}[f(x_{t+1}) \mid x_t] \le \quad & f(x_t) - c\|\nabla f(x_t)\|_{\mathbf{D}}^2 + \frac{1}{n^2}\sum_{j=1}^{n} a^2\lambda_{\mathbf{D}}(f_j(x_k) - f_j^*(x_k)) \\ \le \quad & f(x_t) - c\|\nabla f(x_t)\|_{\mathbf{D}}^2 + \frac{a^2\lambda_{\mathbf{D}}}{n}\left(\frac{1}{n}\sum_{j=1}^{n} f_j(x_t) - \frac{1}{n}\sum_{j=1}^{n} f_j^*\right) \\ \le \quad & f(x_t) - c\|\nabla f(x_t)\|_{\mathbf{D}}^2 + \frac{a^2\lambda_{\mathbf{D}}}{n}(f(x_t) - f^*) + \frac{a^2\lambda_{\mathbf{D}}}{n}\left(f^* - \frac{1}{n}\sum_{j=1}^{n} f_j^*\right).\end{aligned}$$

Substracting f^* from both sides, we get

$$\mathbb{E}[f(x_{t+1}) - f^* \mid x_t] \le \quad f(x_t) - f^* - c\|\nabla f(x_t)\|_{\mathbf{D}}^2 + \frac{a^2\lambda_{\mathbf{D}}}{n}(f(x_t) - f^*) + \frac{a^2\lambda_{\mathbf{D}}}{n}\left(f^* - \frac{1}{n}\sum_{j=1}^{n} f_j^*\right)$$

Now, taking expectation, applying the tower property and rearranging the terms, we get

$$\mathbb{E}[f(x_{t+1}) - f^* \mid x_t] \le \quad \left(1 + \frac{a^2\lambda_{\mathbf{D}}}{n}\right)\mathbb{E}[f(x_t) - f^*] - c\mathbb{E}\left[\|f(x_t)\|_{\mathbf{D}}^2\right] + \frac{a^2\lambda_{\mathbf{D}}}{n}\left(f^* - \frac{1}{n}\sum_{j=1}^{n} f_j^*\right) \tag{44}$$

Say,

$$\zeta_t = \mathbb{E}[f(x_t) - f^*], \quad r_t = \mathbb{E}\left[\|f(x_t)\|_{\mathbf{D}}^2\right], \quad \Delta^* = f^* - \frac{1}{n}\sum_{j=1}^{n} f_j^*,$$

then (44) boils down to

$$cr_t \le \left(1 + \frac{a^2\lambda_{\mathbf{D}}}{n}\right)\zeta_t - \zeta_{t+1} + \frac{a^2\lambda_{\mathbf{D}}}{n}. \tag{45}$$

In order to approach the final result, we now follow (Stich 2019; Khaled and Richtárik 2020)and define an exponentially decaying weighting sequence $\{m_t\}_{t=-1}^{T}$, where T is the total number of iterations. We fix $m_{-1} > 0$ and define

$$m_t = \frac{m_{t-1}}{1 + a^2\lambda_{\mathbf{D}}/n}, \quad \forall \quad t \ge 0.$$

Multiplying both sides of (45) by m_t, we get

$$cm_t r_t \le m_{t-1}\zeta_t - m_t\zeta_{t+1} + \frac{a^2\lambda_D\Delta^*}{n}m_t.$$

Summing up the inequalities from $t = 0, \ldots, T-1$, we get

$$c\sum_{t=1}^{T-1} m_t r_t \le m_{-1}\zeta_0 - m_{T-1}\zeta_T + \frac{a^2\lambda_D\Delta^*}{n}\sum_{t=0}^{t-1} m_t.$$

Define $M_T = \sum_{t=0}^{T-1} m_t$, and divide both sides by M_T, we get

$$c\min_{0\le t\le T-1} r_t \le c\frac{\sum_{t=0}^{T-1} m_t r_t}{M_T} r_t \le \frac{m_{-1}}{M_T}\zeta_0 + \frac{a^2\lambda_D\Delta^*}{n}.$$

Notice that from the definition of m_t, we know that the following inequality holds,

$$\frac{m_{-1}}{M_T} \le \frac{m_{-1}}{Tm_{T-1}} = \frac{(1 + \frac{a^2\lambda_D}{n})^T}{T}.$$

As a result, we have

$$\min_{0\le t\le T-1} r_t \le \frac{\left(1 + \frac{a^2\lambda_D}{n}\right)^T}{cT}\zeta_0 + \frac{a^2\lambda_D\Delta^*}{cn}.$$

Recalling the definition for r_t and ζ_t, we get the following result,

$$\min_{0\leq t\leq T-1}\mathbb{E}\left[\|\nabla f(x_t)\|_{\mathbf{D}}^2\right]\leq\frac{(1+\frac{a^2\lambda_D}{n})^T}{cT}(f(x_0)-f^*)+\frac{a^2\lambda_D\Delta^*}{cn}.$$

Finally, we apply determinant normalization and get

$$\min_{0\leq t\leq T-1}\mathbb{E}\left[\|\nabla f(x_t)\|_{\mathbf{D}/|\mathbf{D}|^{1/d}}^2\right]\leq\frac{(1+\frac{a^2\lambda_D}{n})^T}{c|\mathbf{D}|^{1/d}T}(f(x_0)-f^*)+\frac{a^2\lambda_D\Delta^*}{c|\mathbf{D}|^{1/d}n}. \tag{46}$$

This concludes the proof.

Proof of Corollary 1

Corollary 1. *We reach an ϵ-stationarity, that is the right-hand side of (19) is upper bounded by ϵ^2, if the following conditions are satisfied:*

$$\boldsymbol{DLD}\preceq\boldsymbol{D},\quad\lambda_{\boldsymbol{D}}\leq\min\left\{\frac{n}{T},\frac{cn\epsilon^2}{2\Delta^*}|\boldsymbol{D}|^{1/d}\right\},a^2\leq 1,\quad T\geq\frac{6(f(x_0)-f^*)}{c|\boldsymbol{D}|^{1/d}\epsilon^2}. \tag{47}$$

Proof. Under condition (20), the first term in RHS of (19) can be written as

$$\frac{1}{c}\left(1+\frac{a^2\lambda_{\mathbf{D}}}{n}\right)^T\leq\frac{1}{c}e^{\left(a^2\lambda_{\mathbf{D}}\frac{T}{n}\right)}\leq\frac{1}{c}e\leq\frac{3}{c}$$

Now,

$$\begin{aligned}\frac{\frac{1}{c}\left(1+\frac{a^2\lambda_{\mathbf{D}}}{n}\right)^T(f(x_0)-f^*)}{|\mathbf{D}|^{1/d}T}&\leq\frac{\frac{3}{c}(f(x_0)-f^*)}{|\mathbf{D}|^{1/d}T}\\&\leq\frac{\frac{3}{c}(f(x_0)-f^*)}{|\mathbf{D}|^{1/d}T}\frac{\epsilon^2|\mathbf{D}|^{1/d}}{\frac{6}{c}(f(x_0)-f^*)}\\&\leq\frac{\epsilon^2}{2}.\end{aligned}$$

For the second term in (19), we have

$$\frac{a^2\lambda_{\mathbf{D}}\Delta^*}{c|\mathbf{D}|^{1/d}n}\leq\frac{\Delta^*}{c|\mathbf{D}|^{1/d}n}.\frac{c\epsilon^2|\mathbf{D}|^{1/d}n}{2\Delta^*}\leq\frac{\epsilon^2}{2}.$$

Finally, the LHS of (19) is upper bounded by

$$\min_{0\leq t\leq T-1}\mathbb{E}\left[\|\nabla f(x_t)\|_{\mathbf{D}/|\mathbf{D}|^{1/d}}^2\right]\leq\frac{\epsilon^2}{2}+\frac{\epsilon^2}{2}=\epsilon^2.$$

This concludes the proof. □

Analysis of distributed CFGD-2

Now, we extend CFGD-2 to distributed case. Consider the iterates:

$$x_{t+1}=x_t-\frac{1}{n}\sum_{j=1}^{n}\mathbf{B}_{tj}\mathbf{D}\partial_{b_t}^{\beta,\delta}f_j(x_t), \tag{48}$$

where $\mathbf{D}\in\mathcal{S}_{++}^d$ is the stepsize, and $\mathbf{T}_{tj}$ is a sequence of sketch matrices drawn *i.i.d* from some distributed $\mathcal{B}$ over $\mathcal{S}_+^d$, it satisfies

$$\mathbb{E}[\mathbf{B}_{tj}]=\mathbf{I}_d. \tag{49}$$

Analysis of distributed CFGD-2 (DCFGD-2)

In this section, we present the theory for Algorithm 2, which is an analogous to what we have seen for Algorithm 1. Before proceeding further, we present the following lemma.

Lemma 11. *((Li, Karagulyan, and Richtárik 2023a)) For any sketch matrix $\boldsymbol{B}_{tj}$ of client j randomly drawn from some distribution $\mathcal{B}$ over $\mathcal{S}_+^d$ which satisfies*

$$\mathbb{E}[\boldsymbol{B}_{tj}] = \boldsymbol{I}_d$$

the following holds for any $x \in \mathbb{R}^d$ for each j,

$$\mathbb{E}\left[\|\boldsymbol{B}_{tj}\boldsymbol{D}x - \boldsymbol{D}x\|_{\boldsymbol{L}}^2\right] \leq \lambda_{max}\left(\mathbb{E}\left[\boldsymbol{L}_j^{\frac{1}{2}}\boldsymbol{D}\left(\boldsymbol{B}_{tj} - \boldsymbol{I}_d\right)\boldsymbol{L}\left(\boldsymbol{B}_{tj} - \boldsymbol{I}_d\right)\boldsymbol{D}\boldsymbol{L}_j^{\frac{1}{2}}\right]\right).\|\boldsymbol{x}\|_{\boldsymbol{L}_j^{-1}}^2. \tag{50}$$

Theorem 4. *Let $f_j : \mathbb{R}^d \to \mathbb{R}$ satisfy Assumption 4 and let f satisfy Assumption 1 and 2 with smoothness matrix* $\mathbf{L}$*. Let $\beta \in (0, 1)$. Define $K_j \quad \forall j \in [n]$ as in Lemma 1. If the following conditions are satisfied*

- $\boldsymbol{DLD} \preceq \boldsymbol{D}$
- $|x_t^{(i,j)} - b_t^{(i,j)}| = \mu_j \left|\frac{df}{dx^{(i,j)}}(x_t)\right|, \quad \mu_j \in \left(\frac{-4.236}{K_j}, \frac{0.236}{K_j}\right), \quad \forall i \in [d], j \in [n].$

then, for some $c, a > 0$, the following convergence bound is true for the iterates of CFGD-2:

$$\min_{0 \leq t \leq T-1} \mathbb{E}\left[\|\nabla f(x_t)\|_{\frac{\boldsymbol{D}}{|\boldsymbol{D}|^{1/d}}}^2\right] \leq \frac{\left(1 + \frac{a^2\lambda'_{\boldsymbol{D}}}{n}\right)^T (f(x_0) - f^*)}{c|\boldsymbol{D}|^{1/d}T} + \frac{a^2\lambda'_{\boldsymbol{D}}\Delta^*}{c|\boldsymbol{D}|^{1/d}n}.$$

where $\Delta^* := f^* - \frac{1}{n}\sum_{j=1}^n f_j^*$,

$$\lambda'_{\boldsymbol{D}} := \max_j \left\{\lambda_{max}\left(\mathbb{E}\left[\boldsymbol{L}_j^{\frac{1}{2}}\boldsymbol{D}\left(\boldsymbol{B}_{tj} - \boldsymbol{I}_d\right)\boldsymbol{L}\left(\boldsymbol{B}_{tj} - \boldsymbol{I}_d\right)\boldsymbol{D}\boldsymbol{L}_j^{\frac{1}{2}}\right]\right)\right\}$$

and $a^2 := \max_j \left(\frac{1+K_j\mu_j}{\sqrt{2}}\right)^2$.

Proof. We first define function $g(x)$ as follows,

$$g(x) = \frac{1}{n}\sum_{j=1}^n \mathbf{B}_{tj}\mathbf{D}\partial_{b_t}^{\beta,\delta} f_j(x_t).$$

As a result, Algorithm 2 can be written as

$$x_{t+1} = x_t - g(x_t).$$

Notice that

$$\mathbb{E}[g(x)] = \frac{1}{n}\sum_{j=1}^n \mathbb{E}[\mathbf{B}_{tj}]\mathbf{D}\partial_{b_t}^{\beta,\delta} f_j(x_t) = \mathbf{D}\partial_{b_t}^{\beta,\delta} f(x_t). \tag{51}$$

We then start with the $\mathbf{L}$ matrix smoothness of function f,

$$\begin{aligned}
f(x_{t+1}) &\leq f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t\rangle + \frac{1}{2}\langle \mathbf{L}(x_{t+1} - x_t), x_{t+1} - x_t\rangle \\
&= f(x_t) + \langle \nabla f(x_t), -g(x_t)\rangle + \frac{1}{2}\langle \mathbf{L}(-g(x_t)), -g(x_t)\rangle \\
&= f(x_t) - \langle \nabla f(x_t), g(x_t)\rangle + \frac{1}{2}\langle \mathbf{L}g(x_t), g(x_t)\rangle.
\end{aligned}$$

We then take the expectation conditioned on x_t,

$$\begin{aligned}
\mathbb{E}[f(x_{t+1}) \mid x_t] &\leq f(x_t) - \langle \nabla f(x_t), \mathbb{E}[g(x_t) \mid x_t]\rangle + \frac{1}{2}\mathbb{E}\left[\langle \mathbf{L}g(x_t), g(x_t)\rangle \mid x_t\right] \\
&= f(x_t) - \langle \nabla f(x_t), \mathbf{D}\partial_{b_t}^{\beta,\delta} f(x_t)\rangle + \frac{1}{2}\underbrace{\mathbb{E}\left[\langle \mathbf{L}g(x_t), g(x_t)\rangle \mid x_t\right]}_{:=P}.
\end{aligned} \tag{52}$$

Lemma 6 yields

$$\begin{aligned} P = {} & \mathbb{E}\left[\|g(x_t)\|^2_{\mathbf{L}} \mid x_t\right] \\ = {} & \mathbb{E}\left[\|g(x_t) - \mathbb{E}[g(x_t) \mid x_t]\|^2_{\mathbf{L}} \mid x_t\right] + \|\mathbb{E}[g(x_t) \mid x_t]\|^2_{\mathbf{L}}. \end{aligned}$$

From the unbiasedness of the sketches, we have $\mathbb{E}[g(x_t) \mid x_t] = \mathbf{D}\partial^{\beta,\delta}_{b_t} f(x_t)$, which yields:

$$\begin{aligned} P = {} & \mathbb{E}\left[\|g(x_t) - \mathbf{D}\partial^{\beta,\delta}_{b_t} f_j(x_t)\|^2_{\mathbf{L}} \mid x_t\right] + \|\mathbf{D}\partial^{\beta,\delta}_{b_t} f(x_t)\|^2_{\mathbf{L}} \\ = {} & \mathbb{E}\left[\left\|\frac{1}{n}\sum_{j=1}^{n}\left(\mathbf{B}_{tj}\mathbf{D}\partial^{\beta,\delta}_{b_t} f_j(x_t) - \mathbf{D}\partial^{\beta,\delta}_{b_t} f_j(x_t)\right)\right\|^2_{\mathbf{L}}\middle| x_t\right] + \|\partial^{\beta,\delta}_{b_t} f(x_t)\|^2_{\mathbf{DLD}}. \end{aligned}$$

Using Lemma 7, we have:

$$\begin{aligned} P \overset{(36)}{=} {} & \frac{1}{n^2}\sum_{j=1}^{n}\mathbb{E}\left[\left\|\mathbf{B}_{tj}\mathbf{D}\partial^{\beta,\delta}_{b_t} f_j(x_t) - \mathbf{D}\partial^{\beta,\delta}_{b_t} f_j(x_t)\right\|^2_{\mathbf{L}} |x_t\right] + \|\partial^{\beta,\delta}_{b_t} f(x_t)\|^2_{\mathbf{DLD}} \\ \leq {} & \frac{1}{n^2}\sum_{j=1}^{n}\mathbb{E}\left[\left\|\mathbf{B}_{tj}\mathbf{D}\partial^{\beta,\delta}_{b_t} f_j(x_t) - \mathbf{D}\partial^{\beta,\delta}_{b_t} f_j(x_t)\right\|^2_{\mathbf{L}} |x_t\right] + \|\partial^{\beta,\delta}_{b_t} f(x_t)\|^2_{\mathbf{D}}. \quad \text{(Using the condition } \mathbf{DLD} \preceq \mathbf{D}.\text{)} \end{aligned}$$

By applying Lemma 11

$$\begin{aligned} P \leq {} & \frac{1}{n^2}\sum_{j=1}^{n}\lambda_{max}\left(\mathbb{E}\left[\mathbf{L}_j^{\frac{1}{2}}\mathbf{D}\left(\mathbf{B}_{tj} - \mathbf{I}_d\right)\mathbf{L}\left(\mathbf{B}_{tj} - \mathbf{I}_d\right)\mathbf{D}\mathbf{L}_j^{\frac{1}{2}}\right]\right)\|\partial^{\beta,\delta}_{b_t} f_j(x_t)\|^2_{\mathbf{L}_j^{-1}} + \|\partial^{\beta,\delta}_{b_t} f(x_t)\|^2_{\mathbf{D}} \\ \overset{(9,13)}{\leq} {} & \frac{1}{n^2}\sum_{j=1}^{n}\lambda_{max}\left(\mathbb{E}\left[\mathbf{L}_j^{\frac{1}{2}}\mathbf{D}\left(\mathbf{B}_{tj} - \mathbf{I}_d\right)\mathbf{L}\left(\mathbf{B}_{tj} - \mathbf{I}_d\right)\mathbf{D}\mathbf{L}_j^{\frac{1}{2}}\right]\right)(1 + K_j\mu_j)^2\|\nabla f_j(x_t)\|^2_{\mathbf{L}_j^{-1}} + (1 + K\mu)^2\|\nabla f(x_t)\|^2_{\mathbf{D}} \\ \overset{(38)}{\leq} {} & a^2\lambda'_{\mathbf{D}}\frac{2}{n}\left(f(x_t) - \frac{1}{n}\sum_{j=1}^{n} f_j^*\right) + (1 + K\mu)^2\|\nabla f(x_t)\|^2_{\mathbf{D}} \end{aligned}$$

We plug the upper bound of P back to (52), we get

$$\begin{aligned} \mathbb{E}[f(x_{t+1}) \mid x_t] \leq {} & f(x_t) - \langle\nabla f(x_t), \mathbf{D}\partial^{\beta,\delta}_{b_t} f(x_t)\rangle + \frac{a^2\lambda'_{\mathbf{D}}}{n}\left(f(x_t) - \frac{1}{n}\sum_{j=1}^{n} f_j^*\right) + \frac{(1 + K\mu)^2}{2}\|\nabla f(x_t)\|^2_{\mathbf{D}} \\ \overset{(9,13)}{\leq} {} & f(x_t) - (1 - K\mu)\|\nabla f(x_t)\|^2_{\mathbf{D}} + \frac{a^2\lambda'_{\mathbf{D}}}{n}\left(f(x_t) - \frac{1}{n}\sum_{j=1}^{n} f_j^*\right) + \frac{(1 + K\mu)^2}{2}\|\nabla f(x_t)\|^2_{\mathbf{D}} \\ \leq {} & f(x_t) - \underbrace{\left\{(1 - K\mu) - \frac{(1 + K\mu)^2}{2}\right\}}_{:=c>0 \implies \mu\in\left(\frac{-4.236}{K_j}, \frac{0.236}{K_j}\right)}\|\nabla f(x_t)\|^2_{\mathbf{D}} + \frac{a^2\lambda'_{\mathbf{D}}}{n}\left(f(x_t) - \frac{1}{n}\sum_{j=1}^{n} f_j^*\right) \\ \leq {} & f(x_t) - c\|\nabla f(x_t)\|^2_{\mathbf{D}} + \frac{a^2\lambda'_{\mathbf{D}}}{n}(f(x_t) - f*) + \frac{a^2\lambda'_{\mathbf{D}}}{n}\left(f^* - \frac{1}{n}\sum_{j=1}^{n} f_i^*\right). \end{aligned}$$

Taking expectation, subtracting f^* from both sides, and using the tower property, we get

$$\mathbb{E}\left[f(x_{t+1}) - f^*\right] \leq \mathbb{E}\left[f(x_t) - f^*\right] - c\mathbb{E}\left[\|\nabla f(x_t)\|^2_{\mathbf{D}}\right] + \frac{a^2\lambda_{\mathbf{D}}}{n}\mathbb{E}\left[f(x_t) - f^*\right] + \frac{a^2\lambda_{\mathbf{D}}\Delta^*}{n}.$$

Following similar steps as in the proof of Theorem 2, we are able to get

$$\min_{0\leq t\leq T-1}\mathbb{E}\left[\|\nabla f(x_t)\|^2_{\mathbf{D}}\right] \leq \frac{\left(1 + \frac{a^2\lambda_{\mathbf{D}}}{n}\right)^{\top}}{cT|\mathbf{D}|^{1/d}}\left(f(x_0) - f^*\right) + \frac{a^2\lambda_{\mathbf{D}}\Delta^*}{c|\mathbf{D}|^{1/d}n}. \tag{53}$$

This concludes the proof. □

Similar to Algorithm 1, we can choose the parameters of the algorithm to avoid the exponential blow-up in the convergence bound above. The following corollary sums up the convergence conditions for Algorithm 2.

Corollary 2. *We reach an error level of ϵ^2 in (53) if the following conditions are satisfied:*

$$\boldsymbol{DLD} \preceq \boldsymbol{D}, \quad \lambda'_{\boldsymbol{D}} \leq \min\left\{\frac{n}{T}, \frac{cn\epsilon^2}{2\Delta^*}|\boldsymbol{D}|^{1/d}\right\}, a^2 \leq 1, \quad T \geq \frac{6(f(x_0)-f^*)}{c|\boldsymbol{D}|^{1/d}\epsilon^2}. \tag{54}$$

The proof of this corollary is exactly the same as for Corollary 1.

Optimal stepsise of distributed CFGD-2 (DCFGD-2)

To minimze the iteration complexity of Algorithm 2, the following optimisation problem needs to be solved

$$\begin{array}{ll} \textit{maximize} & \log|\mathbf{D}| \\ \textit{subject to} & \mathbf{D} \quad \text{satisfies} \quad (54). \end{array}$$

One can simply see that the conditions on $\mathbf{D}$ in (54) is convex in nature. One simple way to solve for stepsize matrix $\mathbf{D}$ is to follow the procedure suggested for solving (16). That is, fix $\mathbf{W} \in \mathcal{S}^d_{++}$ and find a real scalar $\omega > 0$ such that $\mathbf{D} = \omega\mathbf{W}$.

D Proofs Of Technical Lemmas

Proof of Lemma 6

Proof. We have

$$\begin{aligned}
\mathbb{E}\left[\|x-\mathbb{E}[x]\|^2_{\mathbf{M}}\right] &= \mathbb{E}\left[(x-\mathbb{E}[x])^\top \mathbf{M}(x-\mathbb{E}[x])\right] \\
&= \mathbb{E}\left[x^\top \mathbf{M}x - \mathbb{E}[x]^\top \mathbf{M}x - x^\top \mathbf{M}\mathbb{E}[x] + \mathbb{E}[x]^\top \mathbf{M}\mathbb{E}[x]\right] \\
&= \mathbb{E}\left[x^\top \mathbf{M}x\right] - 2\mathbb{E}[x]^\top \mathbf{M}\mathbb{E}[x] + \mathbb{E}[x]^\top \mathbf{M}\mathbb{E}[x] \\
&= \mathbb{E}\left[x^\top \mathbf{M}x\right] - \mathbb{E}[x]^\top \mathbf{M}\mathbb{E}[x] \\
&= \mathbb{E}\left[\|x\|^2_{\mathbf{M}}\right] - \|\mathbb{E}[x]\|^2_{\mathbf{M}},
\end{aligned}$$

which concludes the proof. □

Proof of Lemma 7

Proof. We have

$$\begin{aligned}
\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n a_i\right\|^2_{\mathbf{M}}\right] &= \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left[\langle a_i, \mathbf{M}a_i\rangle\right] + \frac{1}{n^2}\sum_{i\neq j}\mathbb{E}\left[\langle a_i, \mathbf{M}a_j\rangle\right] \\
&= \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left[\|a_i\|^2_{\mathbf{M}}\right] + \frac{1}{n^2}\sum_{i\neq j}\langle\mathbb{E}\left[a_i\right], \mathbf{M}\mathbb{E}\left[a_j\right]\rangle \\
&= \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left[\|a_i\|^2_{\mathbf{M}}\right].
\end{aligned}$$

This concludes the proof. □

Proof of Lemma 8

Proof. Notice that

$$\mathbb{E}[\mathbf{S}x] = \mathbb{E}[\mathbf{S}]x = x.$$

We start with variance decomposition in the matrix norm:

$$\begin{aligned}
\mathbb{E}\left[\|Sx-x\|^2_{\mathbf{M}}\right] &\overset{(35)}{=} \mathbb{E}\left[\|\mathbf{S}x\|^2_{\mathbf{M}}\right] - \|x\|^2_{\mathbf{M}} \\
&= \mathbb{E}\left[\langle\mathbf{S}x, \mathbf{MS}x\rangle\right] - \langle x, \mathbf{M}x\rangle \\
&= \langle x, \mathbb{E}[\mathbf{SMS}]x\rangle - \langle x, \mathbf{M}x\rangle \\
&= \langle x, \left(\mathbb{E}[\mathbf{SMS}] - \mathbf{M}\right)x\rangle \\
&= \|x\|^2_{\mathbb{E}[\mathbf{SMS}]-\mathbf{M}} \quad .
\end{aligned}$$

□

Proof of Lemma 9

Proof. We follow the definition of $\mathbf{L}$-matrix smoothness of the function f, that for any $y, x \in \mathbb{R}^d$, we have

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

We plug in $y = x - \mathbf{L}^{-1} \nabla f(x)$, and get

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

Rearranging terms, we get

$$\|\nabla f(x)\|^2_{\mathbf{L}^{-1}} \leq 2\big(f(x) - f^*\big),$$

which completes the proof. □

Proof of Lemma 10

Proof.

$$\begin{aligned}
\mathbb{E}\left[\|\mathbf{G}x - x\|^2_{\mathbf{DLD}}\right] &= \mathbb{E}\left[\langle (\mathbf{G} - \mathbf{I}_d)x, \mathbf{DLD}(\mathbf{G} - \mathbf{I}_d)x \rangle\right] \\
&= x^\top \mathbb{E}[(\mathbf{G} - \mathbf{I}_d)\mathbf{DLD}(\mathbf{G} - \mathbf{I}_d)]x \\
&= x^\top \mathbf{U}^{-\frac{1}{2}} \left(\mathbf{U}^{\frac{1}{2}} \mathbb{E}[(\mathbf{G} - \mathbf{I}_d)\mathbf{DLD}(\mathbf{G} - \mathbf{I}_d)]\mathbf{U}^{\frac{1}{2}}\right) \mathbf{U}^{-\frac{1}{2}} x \\
&\leq \lambda_{\max}\left(\mathbf{U}^{\frac{1}{2}} \mathbb{E}[(\mathbf{G} - \mathbf{I}_d)\mathbf{DLD}(\mathbf{G} - \mathbf{I}_d)]\mathbf{U}^{\frac{1}{2}}\right) \left\|\mathbf{U}^{-\frac{1}{2}} x\right\|^2 \\
&= \lambda_{\max}\left(\mathbf{U}^{\frac{1}{2}} \mathbb{E}[(\mathbf{G} - \mathbf{I}_d)\mathbf{DLD}(\mathbf{G} - \mathbf{I}_d)]\mathbf{U}^{\frac{1}{2}}\right) \|x\|^2_{\mathbf{U}^{-1}}.
\end{aligned}$$

This completes the proof. □

Proof of Lemma 11

Proof.

$$\begin{aligned}
\mathbb{E}\left[\|\mathbf{B}_{tj}\mathbf{D}x - \mathbf{D}x\|^2_{\mathbf{L}}\right] &= \mathbb{E}\left[\langle (\mathbf{B}_{tj}\mathbf{D} - \mathbf{I}_d)x, \mathbf{L}(\mathbf{B}_{tj}\mathbf{D} - \mathbf{I}_d)x \rangle\right] \\
&= x^\top \mathbf{D}\mathbb{E}[(\mathbf{B}_{tj} - \mathbf{I}_d)^\top \mathbf{L}(\mathbf{B}_{tj} - \mathbf{I}_d)]\mathbf{D}x \\
&= x^\top \mathbf{L}_j^{-\frac{1}{2}} \left(\mathbf{L}_j^{\frac{1}{2}} \mathbb{E}[(\mathbf{B}_{tj} - \mathbf{I}_d)^\top \mathbf{L}(\mathbf{B}_{tj} - \mathbf{I}_d)]\mathbf{L}_j^{\frac{1}{2}}\right) \mathbf{L}_j^{-\frac{1}{2}} x \\
&\leq \lambda_{\max}\left(\mathbf{L}^{\frac{1}{2}} \mathbb{E}[(\mathbf{B}_{tj} - \mathbf{I}_d)^\top \mathbf{L}(\mathbf{B}_{tj} - \mathbf{I}_d)]\mathbf{D}\mathbf{L}^{\frac{1}{2}}\right) \|\mathbf{L}_j^{-\frac{1}{2}} x\|^2 \\
&= \lambda_{\max}\left(\mathbf{L}^{\frac{1}{2}} \mathbb{E}[(\mathbf{B}_{tj} - \mathbf{I}_d)^\top \mathbf{L}(\mathbf{B}_{tj} - \mathbf{I}_d)]\mathbf{D}\mathbf{L}^{\frac{1}{2}}\right) \|x\|^2_{\mathbf{L}_j^{-1}}.
\end{aligned}$$

This completes the proof. □

E Experiments

Single node case

For the single node case, we study the logistic regression problem with a non-convex regulariser. The objective is given as

$$f(x) = \frac{1}{n}\sum_{j=1}^{n} \log\left(1 + e^{-b_j \cdot \langle a_j, x\rangle}\right) + \lambda . \sum_{i=1}^{d} \frac{x_i^2}{1 + x_i^2}$$

$x \in \mathbb{R}^d$ represent the model, and let $(a_i, b_i) \in \mathbb{R}^d \times \{-1, +1\}$ denote a single data point from a dataset of size n. The parameter $\lambda > 0$ is a tunable hyperparameter that controls the strength of the regularisation term. For our numerical experiments, we utilize several datasets available in the LibSVM repository (Chang and Lin 2011). The smoothness matrix of f is estimated as follows

$$\mathbf{L} = \frac{1}{n}\sum_{j=1}^{n} \frac{a_j a_j^\top}{4} + 2\lambda . \mathbf{I}_d. \tag{55}$$

Experiments on two-layer neural network

We study the logistic regression problem with a non-convex regulariser using a two-layer neural network (without bias). The objective is given as:

$$f(x_1, x_2) = \frac{1}{n}\sum_{j=1}^{n} \log\left(1 + e^{-b_j \cdot \langle a_j, x_1 + x_2\rangle}\right) + \lambda \cdot \left(\sum_{t=1}^{d} \frac{x_{1t}^2}{1 + x_{1t}^2} + \sum_{t=1}^{d} \frac{x_t^{2t}}{1 + x_{2t}^2}\right) \tag{56}$$

where $x \in \mathbb{R}^d$ is the model, $(a_j, b_j) \in \mathbb{R}^d \times \{-1, +1\}$ is one data point in the dataset whose size is n. The constant $\lambda > 0$ is the hyperparameter associated with the regulariser. We conduct numerical experiments using several datasets from the LibSVM repository (Chang and Lin 2011). We estimate the smoothness matrix of function f here as

$$\mathbf{L} \; = \; \begin{pmatrix} \frac{1}{4n}\sum_{j=1}^{n} a_j a_j^\top \; + \; 2\lambda \, I_d & \frac{1}{4n}\sum_{j=1}^{n} a_j a_j^\top \\ \frac{1}{4n}\sum_{j=1}^{n} a_j a_j^\top & \frac{1}{4n}\sum_{j=1}^{n} a_j a_j^\top \; + \; 2\lambda \, I_d \end{pmatrix}_{2d \times 2d} \tag{57}$$

From Fig 3, we can clearly observe that even in a two-layer network setup, our proposed algorithms (Eqs. CFGD-1 and CFGD-2) outperform both CGD and the state-of-the-art det-CGD family, demonstrating their architecture-agnostic robustness.

Comparison to standard FGD, CFGD with scalar stepsize and CFGD with scalar stepsize and matrix smoothness

The goal of the first experiment is to demonstrate that by employing matrix step sizes, both CFGD-1 and CFGD-2 achieve improved iteration and communication complexities compared to standard FGD and CFGD with scalar stepsize. Specifically, we evaluate FGD and CFGD with a scalar step size ω and scalar smoothness constant $\mathbf{L} = \lambda_{\max}(\mathbf{L})$, as well as CFGD with the step size $\omega \cdot \mathbf{I}_d$ and a smoothness matrix $\mathbf{L}$. In Figure 4, we use the term *standard FGD* and *standard CFGD* to refer to FGD and CFGD respectively with scalar step sizes and scalar smoothness constants, and *CFGD-mat* to refer to CFGD with scalar step sizes and a smoothness matrix.

The metric $G_{T,\mathbf{D}}$, which appears on the y-axis labels in the Figure 4, is defined as:

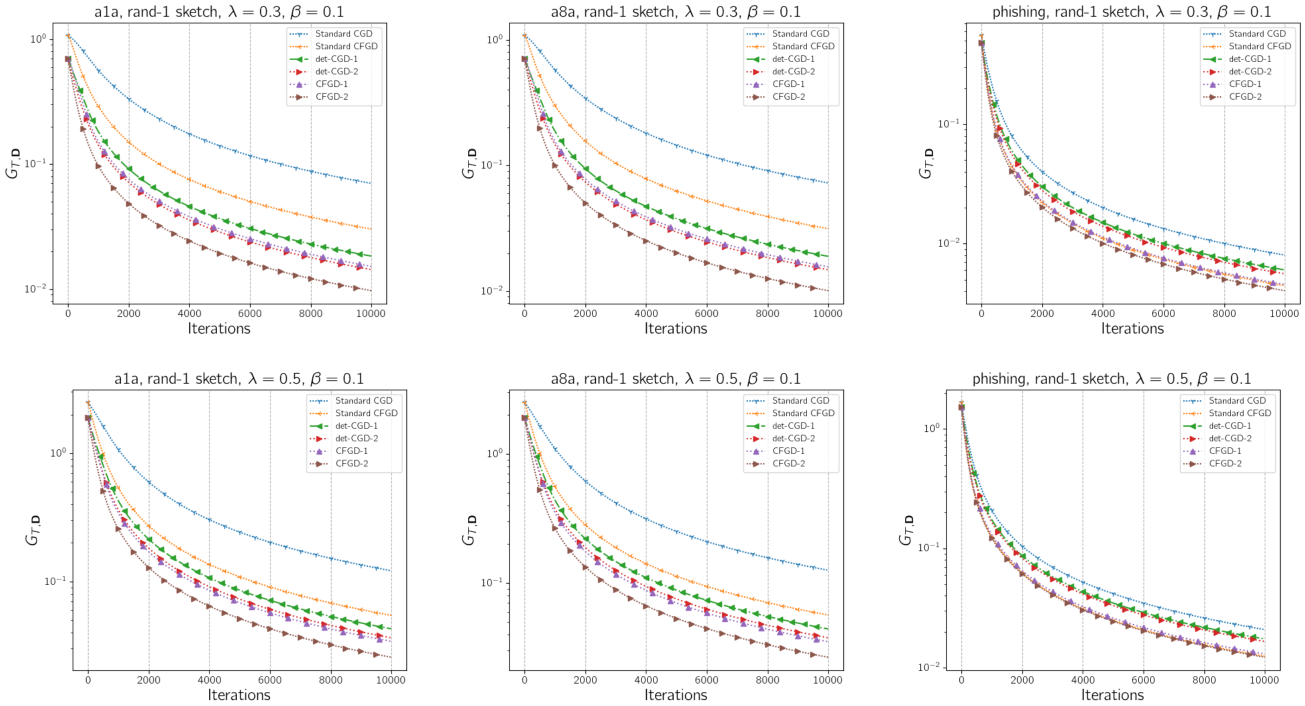
$$G_{T,\mathbf{D}} := \frac{1}{T}\sum_{t=0}^{T-1} \|\nabla f(x_t)\|^2_{\frac{\mathbf{D}}{|\mathbf{D}|^{1/d}}},$$

where $G_{T,\mathbf{D}}$ represents the average matrix norm of the gradient of f over the first T iterations, shown on a logarithmic scale. Here, the weight matrix is scaled to have a determinant of 1, making it comparable to the standard Euclidean norm. This normalization ensures that the results are meaningful.

The outcomes illustrated in Figure 4 indicate that CFGD-mat outperforms standard FGD and CFGD in terms of both iteration complexity and communication complexity. Furthermore, both CFGD-1 and CFGD-2, when using optimal diagonal matrix step sizes, exhibit superior performance compared to standard FGD, CFGD and CFGD-mat, corroborating the theoretical results.

The scaling factors ω_1, ω_2, and ω_3 for CFGD-1 are derived using Theorem 2 with $\ell = 1$. For CFGD-1, the matrix step size is computed according to Equation (17). The experiment also shows that CFGD-1 and CFGD-2 perform similarly when diagonal step sizes are used, which aligns with expectations given that the random-1 sketch ensures commutability between the step size matrix and the sketch matrix, as both are diagonal.

Additionally, CFGD-1 with $\mathbf{D}_2 = \omega_2 . \mathbf{L}^{-1}$ consistently underperforms compared to $\mathbf{D}_4 = \omega_4 . \mathrm{diag}^{-1}(\mathbf{L})$. This behaviour is consistent with the analysis in Section B of the *supplementary material*, where it was noted that row 5 (corresponding to $\mathbf{D}_2$ in Table 1) always yields inferior results compared to row 7 (corresponding to $\mathbf{D}_4$).



a1a, rand-1 sketch, $\lambda = 0.3$, $\beta = 0.1$

a8a, rand-1 sketch, $\lambda = 0.3$, $\beta = 0.1$

phishing, rand-1 sketch, $\lambda = 0.3$, $\beta = 0.1$

a1a, rand-1 sketch, $\lambda = 0.5$, $\beta = 0.1$

a8a, rand-1 sketch, $\lambda = 0.5$, $\beta = 0.1$

phishing, rand-1 sketch, $\lambda = 0.5$, $\beta = 0.1$

Figure 3: Comparison of standard CGD, standard CFGD, det-CGD 1, det-CGD 2 with CFGD-1 and CFGD-2. Throughout the experiments, the Rand-1 sketch is used in all methods. All experiments are performed on a simple two-layer neural network given by the equation 56. All stepsize used are block diagonal approximation of $\mathbf{L}$ given in equation 57. These matrix stepsize are chosen as per **Theorem** 2.

Tuning fractional gradient parameter β

One of the critical hyperparameters for fractional gradient descent is the fractional power $\beta \in (0, 1)$. We conduct an extensive search for β on both the distributed CFGD-2 (DCFGD-2)and simple CFGD-2 (CFGD-2) setups, using the stepsize $\mathbf{D} = \omega.\,\mathrm{diag}^{-1}(\mathbf{L})$. This configuration represents the best-performing algorithm and stepsize combination in both single-node and distributed settings.

From Figure 5, we observe that the algorithms perform well as $\beta \to 0$. However, as $\beta \to 1$, their performance aligns closely with that of simple SGD-based variants, which is intuitive. For instance, the performance of DCFGD-2 with $\beta = 0.9$ is nearly identical to DCGD-2, an SGD variant. This trend remains consistent across both single-node and distributed settings. Additionally, tuning β reveals that the performance of both algorithms does not change significantly, indicating a relatively low dependency on β.

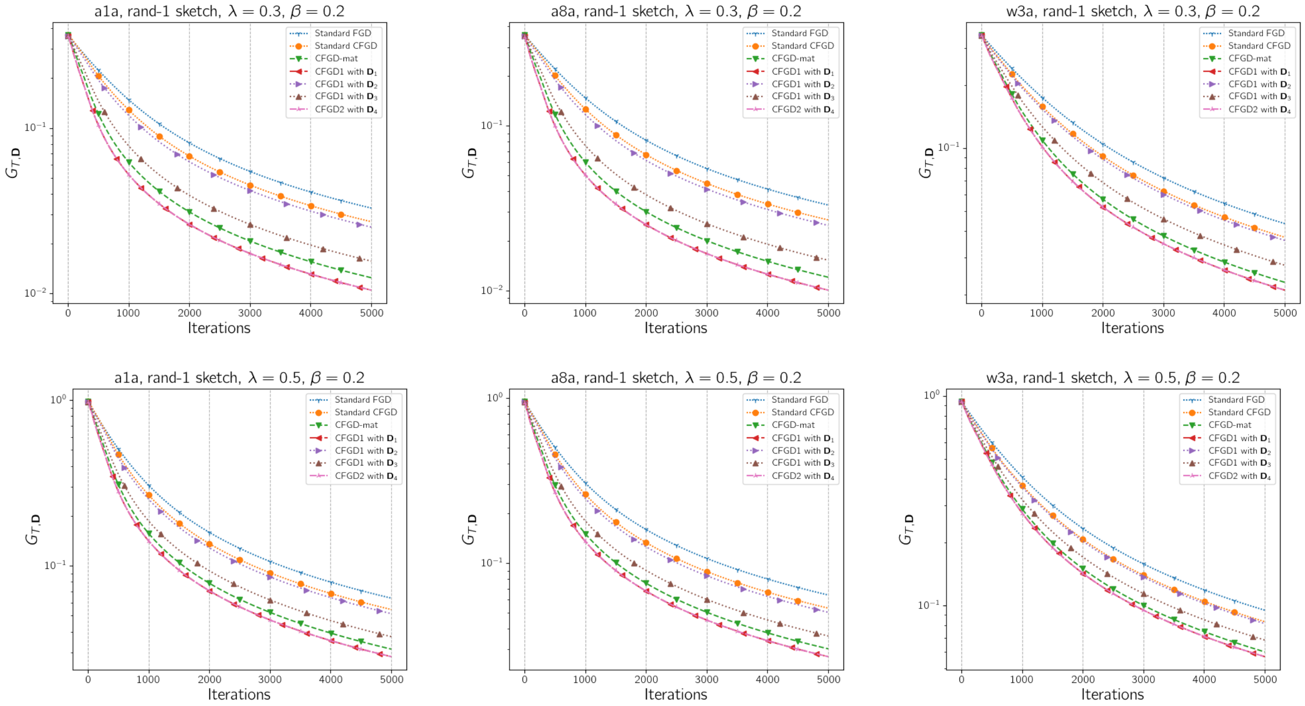


Figure 4: Comparison of standard FGD, standard CFGD, CFGD-mat, CFGD-1 with $\mathbf{D}_1 = \omega_1 \cdot \mathrm{diag}^{-1}(\mathbf{L})$, CFGD-1 with $\mathbf{D}_2 = \omega_2 \cdot \mathbf{L}^{-1}$, CFGD-1 with $\mathbf{D}_3 = \omega_3 \cdot \mathbf{L}^{-1/2}$, and CFGD-2 with $\mathbf{D}_4 = \omega_4 \cdot \mathrm{diag}^{-1}(\mathbf{L})$. Here, $\omega_1, \omega_2, \omega_3$ are the optimal scaling factors for CFGD-1 in their respective cases, and $\mathbf{D}_4$ represents the optimal matrix step size for CFGD-2. Throughout the experiments, the Rand-1 sketch is used in all methods.

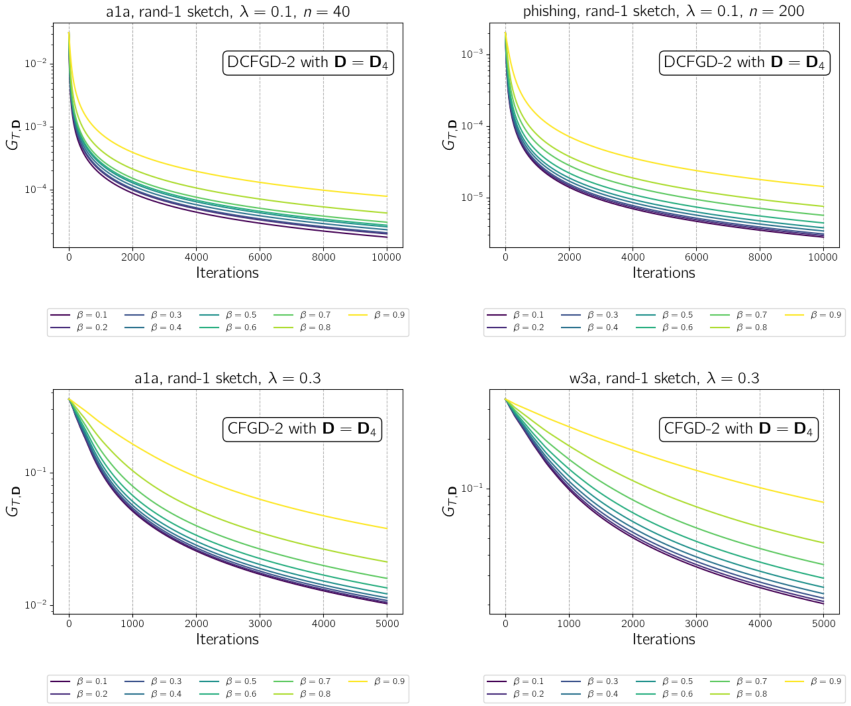


Figure 5: Tuning the fractional gradient parameter β for DCFGD-2 and DCGD-2 with stepsize of type $\mathbf{D}_4 = \omega.\,\mathrm{diag}^{-1}(\mathbf{L})$.