Decentralized Gradient-Free Methods for Stochastic Non-smooth Non-convex Optimization

Zhenwei Lin[1*], **Jingfan Xia**[1*], **Qi Deng**[1†], **Luo Luo**[2]

[1]School of Information Management and Engineering, Shanghai University of Finance and Economics
[2]School of Data Science, Fudan University
zhenweilin@163.sufe.edu.cn, jf.xia@163.sufe.edu.cn, qideng@sufe.edu.cn, luoluo@fudan.edu.cn

Abstract

We consider decentralized gradient-free optimization of minimizing Lipschitz continuous functions that satisfy neither smoothness nor convexity assumption. We propose two novel gradient-free algorithms, the Decentralized Gradient-Free Method (DGFM) and its variant, the Decentralized Gradient-Free Method$^+$ (DGFM$^+$). Based on the techniques of randomized smoothing and gradient tracking, DGFM requires the computation of the zeroth-order oracle of a single sample in each iteration, making it less demanding in terms of computational resources for individual computing nodes. Theoretically, DGFM achieves a complexity of $\mathcal{O}(d^{3/2}\delta^{-1}\varepsilon^{-4})$ for obtaining an (δ,ε)-Goldstein stationary point. DGFM$^+$, an advanced version of DGFM, incorporates variance reduction to further improve the convergence behavior. It samples a mini-batch at each iteration and periodically draws a larger batch of data, which improves the complexity to $\mathcal{O}(d^{3/2}\delta^{-1}\varepsilon^{-3})$. Moreover, experimental results underscore the empirical advantages of our proposed algorithms when applied to real-world datasets.

1 Introduction

In this paper, we consider decentralized optimization where the data are distributed among multiple agents, also known as nodes or entities. For a network with m agents, the optimization problem can be written in the following form:

$$\min_{x\in\mathbb{R}^d} \quad f(x)=\frac{1}{m}\sum_{i=1}^{m} f^i(x), \tag{1}$$

where $f^i(x)=\mathbb{E}_\xi[f^i(x;\xi)]$ is a local cost function on the i-th node and ξ is the index of the random sample. Instead of having a central sever, each node i makes decisions based on its local data and information received from its neighbors. Throughout the paper, we do not require any smoothness or convexity assumption but only suppose that each $f^i(\cdot,\xi)$ is Lipschitz continuous. Moreover, we focus on the gradient-free methods that exclusively rely on function values, avoiding the access for any first-order information.

Decentralized optimization has found extensive applications in signal processing and machine learning (Ling and Tian 2010; Giannakis et al. 2017; Vogels et al. 2021). In the context of smooth non-convex objective that each $f^i(x)$ has the finite-sum structure, a variety of deterministic methods have been proposed (Zeng and Yin 2018; Hong, Hajinezhad, and Zhao 2017; Sun and Hong 2019; Scutari and Sun 2019; Xin, Khan, and Kar 2022; Luo and Ye 2022). Notably, Xin, Khan, and Kar (2022) achieved a network topology-independent convergence rate in a big-data regime. Luo and Ye (2022) integrated variance reduction, gradient tracking, and multi-consensus techniques, yielding an algorithm that meets the tighter communication requirement and complexity level of first-order oracle algorithms. In the realm of stochastic decentralized optimization, a significant body of literature has explored acceleration techniques that incorporate variance reduction (Pan, Liu, and Wang 2020; Sun, Lu, and Hong 2020; Xin, Khan, and Kar 2021a,b).

Moreover, a substantial volume of literature exists regarding resolving non-convex non-smooth optimization (Di Lorenzo and Scutari 2016; Scutari and Sun 2019; Wang et al. 2021; Xin et al. 2021; Mancino-Ball et al. 2023; Xiao et al. 2023; Chen, Garcia, and Shahrampour 2021; Wang et al. 2023). However, most existing research require the objective to adhere to a specific structure. Predominantly, studies focus on composite optimization, where the objective sums up a smooth non-convex part and a possibly non-smooth part. In this vein, Scutari and Sun (2019) introduced a decentralized algorithmic framework for minimization of the sum of a smooth non-convex function and a non-smooth difference-of-convex function over a time-varying directed graph. Mancino-Ball et al. (2023) introduced a single-loop algorithm with a small batch size which achieved a network topology-independent complexity. Conversely, other recent investigations have focused on the decentralized optimization of non-smooth weakly-convex functions (Chen, Garcia, and Shahrampour 2021; Wang et al. 2023).

Previous decentralized algorithms still require gradient computation, while this oracle may be computationally prohibitive (Liu et al. 2020), such as sensor selection (Liu et al. 2018). Moreover, the gradient-free method has a promising application in adversarial machine learning, especially in black-box adversarial attacks (Chen, Jordan, and Wainwright 2020; Moosavi-Dezfooli et al. 2017). Recently, some works studied decentralized optimization problem with zeroth-order methods (Tang, Zhang, and Li 2020;

[*]These authors contributed equally.
[†]Corresponding author.

Sahu et al. 2018; Yu, Ho, and Yuan 2021; Hajinezhad, Hong, and Garcia 2019; Tang, Ren, and Li 2023). For example, Sahu et al. (2018) considered convex problems, while Hajinezhad, Hong, and Garcia (2019) focused on non-convex problems. Moreover, some attention has been paid to applying zero-order decentralized algorithms to constrained optimization (Yu, Ho, and Yuan 2021; Tang, Ren, and Li 2023). However, these researches only focus on smooth optimization.

The decentralized algorithms described above can be classified into smooth non-convex and non-smooth non-convex with specific structures. This leads us to raise the following question: *Can we develop a decentralized gradient-free algorithm that has provable complexity guarantees for non-smooth, non-convex but Lipschitz continuous problems?* To address this research problem, a natural idea is to extend the centralized algorithms designed for non-smooth non-convex problems to the decentralized setting. Zhang et al. (2020) introduced (δ, ε)-Goldstein stationarity as a valid criterion for non-smooth non-convex optimization, which makes it possible to analyze the non-asymptotic convergence. They utilize a random sampling approach to choose an interpolation point on the segment connecting two iterates. This method guarantees a substantial descent of the objective function, given the assumption that the function is Hadamard directionally differentiable and access to a generalized gradient oracle is available. This algorithm can achieve complexity $\mathcal{O}(\Delta L_f^3 \delta^{-1} \varepsilon^{-4})$, where L_f is the Lipschitz continuous constant of the objective and Δ is the inital function value gap. Later, Davis et al. (2022) and Tian, Zhou, and So (2022) relaxed the subgradient selection oracle assumption and Hadamard directionally differentiable assumption by adding random perturbation. More recently, Cutkosky, Mehta, and Orabona (2023) found a connection between non-convex stochastic optimization and online learning and established a stochastic first-order oracle complexity of $\mathcal{O}(\Delta L_f^2 \delta^{-1} \varepsilon^{-3})$, which is the optimal in the case of $\varepsilon \leq \mathcal{O}(\delta)$. In light of recent advances in designing zeroth-order algorithms for non-smooth non-convex problems, an effective way is by applying the randomized smoothing technique (Nesterov and Spokoiny 2017; Shamir 2017). This approach constructs a smooth surrogate function to which algorithms for smooth functions can be applied. Lin, Zheng, and Jordan (2022) established the relationship between Goldstein stationarity of the original objective function and ε-stationarity of the surrogate function, and presented an algorithm for finding a (δ, ε)-Goldstein stationary point within at most $\mathcal{O}(d^{3/2} L_f^4 \varepsilon^{-4} + d^{3/2} \Delta L_f^3 \delta^{-1} \varepsilon^{-4})$ stochastic zeroth-order oracle calls. Later, Chen, Xu, and Luo (2023) constructed stochastic recursive gradient estimators to accelerate and achieve a stochastic zeroth-order oracle complexity of $\mathcal{O}(d^{3/2} L_f^3 \varepsilon^{-3} + d^{3/2} \Delta L_f^2 \delta^{-1} \varepsilon^{-3})$. All above work applied the first-order or zero-order methods for finding the approximate stationary point of the smooth surrogate function. Very recently, Kornowski and Shamir (2023) replaced the goal of finding an ε-stationary point of the smoothed function with that of finding a Goldstein stationary point and then used a stochastic first-order non-smooth non-convex algorithm. This change led to an improved dependence on the dimension, reducing it from $d^{3/2}$ to d.

1.1 Contributions

In this work, we propose two gradient-free decentralized algorithms for non-smooth non-convex optimization: the Decentralized Gradient Free Method (DGFM) and the Decentralized Gradient Free Method$^+$ (DGFM$^+$).

DGFM is a decentralized approach that leverages randomized smoothing and gradient tracking techniques. DGFM only requires the computation of the zeroth-order oracle of a single sample, thus reducing the computational demands on individual computing nodes and enhancing practicality. Theoretically, DGFM achieves a complexity bound of $\mathcal{O}(d^{3/2} \delta^{-1} \varepsilon^{-4})$ for reaching a (δ, ε)-Goldstein stationary solution in expectation. Furthermore, DGFM requires the same number of communication rounds as the number of iterations. To the best of our knowledge, DGFM is the first decentralized algorithm for general non-smooth non-convex optimization problems. Our complexity result matches the complexity bound of the standard gradient-free stochastic method (Lin, Zheng, and Jordan 2022).

We also propose an enhanced algorithm DGFM$^+$ by incorporating the variance reduction technique SPIDER (Fang et al. 2018). DGFM$^+$ samples a mini-batch at each iteration and periodically samples a mega-batch of data. This strategy improve the zeroth-order oracle complexity to $\mathcal{O}(d^{3/2} \delta^{-1} \varepsilon^{-3})$ for reaching (δ, ε)-Goldstein stationarity in expectation. In comparison to DGFM, DGFM$^+$ not only employs randomized smoothing and gradient tracking but also introduces a multi-communication module. This addition ensures that the order of communication complexity is on par with that of the iteration complexity.

2 Preliminaries

We give some notations and introduce basic concepts in non-smooth and non-convex analysis.

Notations. We use subscripts to indicate the nodes to which the variables belong and superscripts to indicate the iteration numbers. We use bold letters such as $\mathbf{x}$ to represent stack variables e.g., $\mathbf{x}^k = [(x_1^k)^\top, \cdots, (x_m^k)^\top]^\top \in \mathbb{R}^{md}$. We denote $\|x\|_q = (\sum_{i=1}^d |x_{(i)}|^q)^{1/q}$, $q > 0$ for the ℓ_q-norm, where $x_{(i)}$ denote the i-th element of x. For brevity, $\|x\|$ stands for ℓ_2-norm. For a matrix A, we denote its spectral norm as $\|A\|$. The notation $\mathcal{B}_\nu(x)$ presents a closed Euclidean ball centered at x with radius $\nu > 0$, i.e., $\mathcal{B}_\nu(x) = \{y : \|y - x\| \leq \nu\}$. We use $\mathbb{S}^{d-1} = \{x \in \mathbb{R}^d : \|x\| = 1\}$ to denote the sphere of the unit ball in ℓ_2-norm. We work with a probability space $\{\Omega, \mathcal{F}, \mathbb{P}\}$, where Ω is a sample space containing all possible outcomes, $\mathcal{F}$ is the sigma-algebra on Ω representing the set of events, and $\mathbb{P}$ is the probability measure. In this paper, we use $\mathbb{P}^d$ to denote the uniform distribution on $\mathbb{S}^{d-1}$. We consider decentralized algorithms that generate a sequence $\{x_i^k\}_{k \geq 0}$ to approximate the stationary point of $f(\cdot)$. At each iteration k, each node i observes a random vector set $S_i^k = \{(\xi_i^{k,j}, w_i^{k,j})\}_{j=1}^b$,

where ξ is the data sample, w is a random vector sample for gradient estimation and b is the batch size. We introduce a natural filtration induced by these random vector sets observed sequentially by these nodes: $\mathcal{F}_0 := \{\Omega, \varnothing\}$ and $\mathcal{F}_k := \sigma(\{S_i^0, S_i^1, \cdots, S_i^{k-1} : i \in [m]\})$ for any $k \geq 1$. We use $\mathbb{E}\left[\left\|\mathbf{x}_\perp^k\right\|^2\right] := \mathbb{E}\left[\left\|\mathbf{x}^k - \bar{\mathbf{x}}^k\right\|^2\right]$ to denote consensus error of sequence $\{x_k\}$, and the consensus error under $\mathcal{F}_k$ are denoted by $\mathbb{E}\left[\left\|\mathbf{x}_\perp^k\right\|^2 \mid \mathcal{F}_k\right] := \mathbb{E}\left[\left\|\mathbf{x}^k - \bar{\mathbf{x}}^k\right\|^2 \mid \mathcal{F}_k\right]$. Furthermore, similar notations hold for y, which is a gradient tracking vector of stochastic gradient.

Stationary condition. In the non-convex setting, Clarke's subdifferential (or generalized gradient) (Clarke 1990) is perhaps the most natural and well-known extension of the standard convex subdifferential.

Definition 2.1. *Given a point $x \in \mathbb{R}^d$ and the direction $v \in \mathbb{R}^d$, the generalized directional derivative of a Lipschitz continuous function f is given by*

$$Df(x; v) := \limsup_{y \to x, t \downarrow 0} \frac{f(y + tv) - f(y)}{t}.$$

The generalized gradient of f is defined as

$$\partial f(x) := \{g \in \mathbb{R}^d : g^\top v \leq Df(x; v), \forall v \in \mathbb{R}^d\}.$$

We need to properly define the approximate stationary condition for the efficiency analysis. An intuitive choice is to consider the ε-Clarke's stationary point which is defined by $\text{dist}(0, \partial f(x)) \leq \varepsilon$. However, Kornowski and Shamir (2021) demonstrated that accessing such approximate stationarity for sufficiently small ε tends to be generally intractable. Therefore, as suggested by Lin, Zheng, and Jordan (2022); Zhang et al. (2020), it is more sensible to target a (δ, ε)-Goldstein stationary point (Goldstein 1977).

Definition 2.2 ((δ, ε)-Goldstein Stationary Point)**.** *Given $\delta > 0$, the δ-Goldstein subdifferential of Lipschitz function $f(\cdot)$ at x is given by $\partial_\delta f(x) := \text{conv}(\cup_{y \in \mathbb{B}_\delta(x)} \partial f(y))$. Then we say point x is a (δ, ε)-Goldstein stationary point if $\min\{\|g\| : g \in \partial_\delta f(x)\} \leq \varepsilon$.*

Randomized Smoothing. For non-smooth problems, a natural idea is first to apply smoothing techniques to these problems and then minimize the resulting smoothed surrogate function. We highlight some key properties and refer to (Lin, Zheng, and Jordan 2022) for more details.

Definition 2.3 (Randomized smoothing)**.** *We say function $f_\delta(x)$ is a randomized smooth approximation of the non-smooth function $f(x)$ if $f_\delta(x) := \mathbb{E}_{w \sim \mathbb{P}}[f(x + \delta \cdot w)]$.*

The randomized smooth approximation requires the objective function $f(x) = \frac{1}{m}\sum_{i=1}^m \mathbb{E}_\xi[f^i(x; \xi)]$ to satisfy the following assumption to have good properties.

Assumption 1. *For $\forall i \in [m]$, assume condition $\left\|f^i(x; \xi) - f^i(y; \xi)\right\| \leq L(\xi)\|x - y\|$ holds, namely, $f^i(\cdot\,; \xi)$ is $L(\xi)$-Lipschitz continuous. Furthermore, assume there exists a constant L_f such that $\mathbb{E}[L(\xi)^2] \leq L_f^2$. Moreover, we assume $f(\cdot)$ is lower bounded and define $f^* = \inf_{x \in \mathbb{R}^d} f(x)$.*

Then we give a specific gradient-free method to approximate the gradient.

Definition 2.4 (Zeroth-order oracle estimators)**.** *Given a stochastic component $f(\cdot; \xi) : \mathbb{R}^d \to \mathbb{R}$, we define its zeroth-order oracle estimator at $x \in \mathbb{R}^d$ by:*

$$g(x; w, \xi) = \frac{d}{2\delta}(f(x + \delta \cdot w; \xi) - f(x - \delta \cdot w; \xi))w,$$

where w is uniformly sampled from $\mathbb{S}^{d-1}$. Let $S = \{(\xi_i, w_i)\}_{i=1}^b$, where vectors $w_1, \cdots, w_b \in \mathbb{R}^d$ are i.i.d sampled from $\mathbb{S}^{d-1}$ and random indices $\xi_1, \cdots, \xi_b$ are i.i.d. We define the mini-batch zeroth-order gradient estimator of $f(\cdot; \xi)$ in terms of S at $x \in \mathbb{R}^d$ by

$$g(x; S) = \frac{1}{b}\sum_{i=1}^b g(x; w^i, \xi^i).$$

Proposition 2.1 (Lemma D.1 (Lin, Zheng, and Jordan 2022))**.** *For the zeroth-order oracle estimator in Definition 2.4, we have $\mathbb{E}_{w,\xi}[g(x; w, \xi)] = \nabla f_\delta(x)$ and $\mathbb{E}_{w,\xi}\left[\left\|g(x; w, \xi)\right\|^2\right] \leq 16\sqrt{2\pi} d L_f^2$.*

In the remains of this paper, we use the notation $\sigma^2 = 16\sqrt{2\pi} d L_f^2$.

We summarize the main properties of randomized smoothing in the following proposition.

Proposition 2.2 (Proposition 2.2 (Chen, Xu, and Luo 2023))**.** *Suppose Assumption 1 holds, then for the local function $f^i(x) = \mathbb{E}[f^i(x; \xi)]$, we have*

1. *$|f^i(\cdot) - f_\delta^i(\cdot)| \leq \delta L_f$.*
2. *$f_\delta^i(x)$ is L_f-Lipschitz continuous and L_δ-smooth, where $L_\delta = cL_f\sqrt{d}/\delta$ and c is a constant.*
3. *$\nabla f_\delta^i(\cdot) \in \partial_\delta f^i(\cdot)$.*
4. *There exists Δ such that for any $x \in \mathbb{R}^d$, $f_\delta(x) - f^* \leq \Delta_\delta$, where $\Delta_\delta = \Delta + \delta L_f$.*

Graph. Consider a time-varying network $(\mathcal{V}, \mathcal{E}_k)$ of agents, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}_k$ is the set of links connecting nodes at time k. Let $A(k) \triangleq (a_{i,j}(k))_{i,j=1}^m$ denote the matrix of weights associated with links in the graph at time $k > 0$. In addition, we will use $A^\tau(k) \triangleq (a_{i,j}^\tau(k))_{i,j=1}^m$ to denote the weighted matrix (mixing matrix) for the τ-th repetition of communication in k-th iterations. We make the following standard assumption on the graph topology.

Assumption 2 (Graph property)**.** *The weighted matrix $A(k)$ has a sparsity pattern compliant with $\mathcal{G}$ that is*

1. *$a_{i,i}(k) > 0$, for any i and k;*
2. *$a_{i,j}(k) > 0$ if $(i, j) \in \mathcal{E}_k$ and $a_{i,j}(k) = 0$ otherwise;*
3. *$A(k)$ is doubly stochastic, which means $\mathbf{1}^\top A(k) = \mathbf{1}^\top$ and $A(k)\mathbf{1} = \mathbf{1}$*

Furthermore, all of above properties hold if we replace $A(k)$ with $A^\tau(k)$.

Remark 1. *Several rules for selecting weights for local averaging have been proposed in the literature that satisfies Assumption 2 (Xiao, Boyd, and Lall 2005). Examples include the Laplacian, the Metropolis-Hasting, and the maximum-degree weights rules.*

The doubly stochastic matrix has some desirable properties (Tsitsiklis 1984; Sun, Scutari, and Daneshmand 2022) that will be used in our analysis.

Proposition 2.3. *Let* $\tilde{A}(k) = A(k) \otimes \mathbf{I}_d$, $J = \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^\top \otimes \mathbf{I}_d$, *then:*

1. $\tilde{A}(k)J = J = J\tilde{A}(k)$.
2. $\tilde{A}(k)\bar{\mathbf{z}}^k = \bar{\mathbf{z}}^k = J\bar{\mathbf{z}}^k, \forall \bar{z}^k \in \mathbb{R}^d$.
3. *There exists* ρ *such that* $\max_k \left\{\left\|\tilde{A}(k) - J\right\|\right\} \leq \rho < 1$.
4. $\left\|\tilde{A}(k)\right\| \leq 1$.

We remark that Proposition 2.3 also hold by replacing $A(k)$ with $A^\tau(k)$.

Algorithm 1: DGFM at each node i

Require: $x_i^{-1} = x_i^0 = \bar{x}^0, \forall i \in [m]$, K, η

Initialize: $y_i^0 = g_i(x_i^{-1}; S_i^{-1}) = \mathbf{0}_d$

for $k = 0, \ldots, K-1$ **do**

  Sample $S_i^k = \left\{\xi_i^{k,1}, w_i^{k,1}\right\}$ and calculate $g_i(x_i^k; S_i^k)$

  $y_i^{k+1} = \sum_{j=1}^m a_{i,j}(k)(y_j^k + g_j(x_j^k; S_j^k) - g_j(x_j^{k-1}; S_j^{k-1}))$

  $x_i^{k+1} = \sum_{j=1}^m a_{i,j}(k)(x_j^k - \eta y_j^{k+1})$

Return: Choose x_{out} uniformly at random from $\left\{x_i^k\right\}_{k=1,\cdots,K, i=1,\cdots,m}$

3 DGFM

In this section, we develop the decentralized Gradient-Free Method (DGFM), an extension of the centralized zeroth-order method proposed by Lin, Zheng, and Jordan (2022). In a multi-agent environment, a significant challenge lies in managing the consensus error among agents to match the order of the optimization error. To address this issue, DGFM integrates a widely-used gradient tracking technique (Di Lorenzo and Scutari 2016; Sun, Scutari, and Daneshmand 2022; Nedic, Olshevsky, and Shi 2017; Lu et al. 2019) with the gradient-free method. Specifically, for every node i, DGFM contains the following three steps. Firstly, it sample a random direction w_i^k and a data point ξ_i^k, and calculate the corresponding zeroth-order oracle estimator $g_i(x_i^k; S_i^k)$, where $S_i^k = (w_i^k, \xi_i^k)$. Secondly, the gradient tracking technique is applied to monitor the zeroth-order oracle estimator of the overall function. Lastly, the primal variable is updated using a perturbed and locally weighted average. We present the details in Algorithm 1.

We first establish some basic properties of the ergodic sequences, especially for the average sequences.

Lemma 3.1. *Let* $\{x_i^k, y_i^k, g_i(x_i^k; S_i^k)\}$ *be the sequence generated by DGFM and* $\{\mathbf{x}^k, \mathbf{y}^k, \mathbf{g}^k\}$ *be the corresponding stack variables, then we have*

$$\mathbf{x}^{k+1} = \tilde{A}(k)(\mathbf{x}^k - \eta\mathbf{y}^{k+1}), \bar{\mathbf{x}}^{k+1} = \bar{\mathbf{x}}^k - \eta\bar{\mathbf{y}}^{k+1},$$
$$\bar{\mathbf{y}}^{k+1} = \bar{\mathbf{y}}^k + \bar{\mathbf{g}}^k - \bar{\mathbf{g}}^{k-1},$$
$$\bar{\mathbf{y}}^{k+1} = \bar{\mathbf{g}}^k \quad \textit{and} \quad \mathbf{y}^{k+1} = \tilde{A}(k)(\mathbf{y}^k + \mathbf{g}^k - \mathbf{g}^{k-1}),$$

where $\bar{x}^k = \frac{1}{m}\sum_{i=1}^m x_i^k, \bar{\mathbf{x}}^k = \mathbf{1}_m \otimes \bar{x}^k, \bar{y}^k = \frac{1}{m}\sum_{i=1}^m y_i^k$ *and* $\bar{\mathbf{y}}^k = \mathbf{1}_m \otimes \bar{y}^k$.

Lemma 3.1 implies that DGFM effectively executes an approximate gradient descent on the consensus sequence, utilizing the variable y to track gradients for approximating the overall gradient. Subsequently, we present the main convergence results of DGFM and discuss the relevant features. The result of consensus error decay at exponential order is given in Lemma 3.2.

Lemma 3.2 (Consensus error decay)**.** *Let* $\{x_i^k, y_i^k, g_i(x_i^k; S_i^k)\}$ *be the sequence generated by DGFM and* $\{\mathbf{x}^k, \mathbf{y}^k, \mathbf{g}^k\}$ *be the corresponding stack variables, then for* $k \geq 0$*, there exist positive* α_1 *and* α_2 *such that*

$$\mathbb{E}\left[\left\|\mathbf{x}_\perp^{k+1}\right\|^2\right] \leq \rho^2(1+\alpha_1)\mathbb{E}\left[\left\|\mathbf{x}_\perp^k\right\|^2\right] + \rho^2\eta^2(1+\alpha_1^{-1})\mathbb{E}\left[\left\|\mathbf{y}_\perp^{k+1}\right\|^2\right],$$

and

$$\mathbb{E}\left[\left\|\mathbf{y}_\perp^{k+1}\right\|^2 \mid \mathcal{F}_k\right] - \rho^2(1+\alpha_2)\mathbb{E}\left[\left\|\mathbf{y}_\perp^k\right\|^2 \mid \mathcal{F}_k\right] \leq \rho^2(1+\alpha_2^{-1})(6+36\eta^2L_\delta^2)m\sigma^2 + \theta_3\mathbb{E}\left[\left\|\mathbf{x}_\perp^{k-1}\right\|^2 \mid \mathcal{F}_k\right] + \theta_2\mathbb{E}\left[\left\|\mathbf{x}_\perp^k\right\|^2 \mid \mathcal{F}_k\right] + \theta_1\mathbb{E}\left[\left\|\nabla f_\delta(\bar{x}^{k-1})\right\|^2 \mid \mathcal{F}_k\right],$$

where $\theta_1 = 18L_\delta^2\eta^2\rho^2(1+\alpha_2^{-1})$, $\theta_2 = 9\rho^2(1+\alpha_2^{-1})L_\delta^2$, $\theta_3 = 9\rho^2(1+\alpha_2^{-1}L_\delta^2(1+4\eta^2L_\delta^2)$ *and* $L_\delta = cL_f\sqrt{d}\delta^{-1}$ *is the smoothness of* $f_\delta(\cdot)$.

Lemma 3.2 indicates that if we omit some additional error term other than $\mathbb{E}\left[\left\|\mathbf{x}_\perp^k\right\|^2\right]$ and $\mathbb{E}\left[\left\|\mathbf{y}_\perp^k\right\|^2\right]$, there is a factor $\rho^2(1+\alpha_i), i = 1, 2$, between two successive consensus error terms. Since ρ is smaller than 1, we can choose a suitable α_i (such as $(1-\rho^2)/(2\rho^2)$) so that the factor $\rho^2(1+\alpha_i) < 1$, thereby achieving an exponential decrease in the consensus error. Therefore, we can show the descent property of $\left\{\bar{x}^k\right\}$ in the following Lemma 3.3 and combine it with Lemma 3.2 to get the descent property of the overall sequence in Lemma 3.4.

Lemma 3.3. *Let* $\{\mathbf{x}^k, \mathbf{y}^k, \bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k\}$ *be the sequence generated by DGFM, then we have*

$$\mathbb{E}\left[f_\delta(\bar{x}^{k+1})\right] - \mathbb{E}\left[f_\delta(\bar{x}^k)\right] + \frac{\eta}{2}\mathbb{E}\left[\left\|\nabla f_\delta(\bar{x}^k)\right\|^2\right] \leq \frac{\eta L_\delta^2}{2m}\mathbb{E}\left[\left\|\mathbf{x}_\perp^k\right\|^2\right] + L_\delta\eta^2(\sigma^2 + L_f^2).$$

We obtain Lemma 3.4 by multiplying the two consensus errors in Lemma 3.2 by their corresponding factors β_x and β_y and adding them to the result of Lemma 3.3.

Lemma 3.4 (Informal)**.** *Let* $\{\mathbf{x}^k, \mathbf{y}^k, \bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k\}$ *be the sequence generated by DGFM, then there exist positive constants* β_x *and* β_y *such that*

$$\theta_4\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla f_\delta(\bar{x}^k)\right\|^2\right] + (\theta_5 - \theta_6)\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\mathbf{x}_\perp^k\right\|^2\right] \leq \mathbb{E}\left[f_\delta(\bar{x}^0)\right] - \mathbb{E}[f_\delta(\bar{x}^K)] + (\theta_6 - \beta_x)\mathbb{E}\left[\left\|\mathbf{x}_\perp^K\right\|^2\right] + \theta_8 - \theta_7\sum_{k=1}^K\mathbb{E}\left[\left\|\mathbf{y}_\perp^k\right\|^2\right] - \beta_y(\mathbb{E}\left[\left\|\mathbf{y}_\perp^{K+1}\right\|^2\right] - \mathbb{E}\left[\left\|\mathbf{y}_\perp^1\right\|^2\right]),$$

where constants $\theta_4, \theta_5, \theta_6, \theta_7$ *and* θ_8 *depend on* $\alpha_1, \alpha_2, \beta_x, \beta_y, d, \eta, L_\delta, L_f, m$ *and* ρ.

Observe that Lemma 3.4 can give an upper-bound of $\mathbb{E}\big[\big\|\nabla f_\delta(\bar{x}^k)\big\|^2\big] + (\theta_5-\theta_6)\theta_4{}^{-1}\mathbb{E}\big[\big\|x^k-\bar{x}^k\big\|^2\big]$ when we choose the proper parameters of $\alpha_1,\alpha_2,\beta_x,\beta_y,\eta$ and δ such that $\theta_i > 0$ for $i = 4,\cdots,8$ and it holds that $(\theta_5-\theta_6)\theta_4{}^{-1} = \mathcal{O}(\delta^{-2})$. Since $\mathbb{E}\big[\big\|\nabla f_\delta(x_i^k)\big\|^2\big] \le 2L_\delta^2\mathbb{E}\big[\big\|x_i^k-\bar{x}^k\big\|^2\big]+2\mathbb{E}\big[\big\|\nabla f_\delta(\bar{x}^k)\big\|^2\big]$, then we can give an upper bound of $\mathbb{E}\big[\big\|\nabla f_\delta(x_i^k)\big\|^2\big]$, which naturally leads to the complexity results of finding (δ,ε)-stationary points with simple calculation. Next, we present more specific parameter settings, leading to the optimal convergence rate.

Theorem 3.1 (Informal)**.** *DGFM outputs a (δ,ε)-Goldstein stationary point of $f(\cdot)$ in expectation with the total stochastic zeroth-order complexity and total communication complexity at most $\mathcal{O}(\Delta_\delta\delta^{-1}\varepsilon^{-4}d^{3/2})$ by setting $\alpha_1=\alpha_2=(1-\rho^2)/2\rho^2$, $\delta=\mathcal{O}(\varepsilon)$, $\beta_x=\mathcal{O}(\delta^{-1}),\beta_y=\mathcal{O}(\varepsilon^4\delta)$ and $\eta=\mathcal{O}(\varepsilon^2\delta)$.*

4 DGFM$^+$

In this section, we consider DGFM$^+$. First, we give some preliminaries in the following section.

Algorithm 2: DGFM$^+$ at each node i

Require: $x_i^{-1}=x_i^0=\bar{x}^0,y_i^0=v_i^{-1}=\mathbf{0}_d,\forall i\in[m]$, K,η,b,b'

for $k=0,\ldots,K-1$ **do**
  if k mod $T=0$ **then**
    Sample $S_i^{k\prime}=\big\{(\xi_i^{k\prime,j},w_i^{k\prime,j})\big\}_{j=1}^{b'}$
    Calculate $y_i^{k+1}=v_i^k=g_i(x_i^k;S_i^k)$
    for $\tau=1,\cdots,\mathcal{T}$ **do**
      $y_i^{k+1}=\sum_{j=1}^m a_{i,j}^\tau(k)y_j^{k+1}$
  else
    Sample $S_i^k=\big\{(\xi_i^{k,j},w_i^{k,j})\big\}_{j=1}^{b}$
    $v_i^k=v_i^{k-1}+g_i(x_i^k;S_i^k)-g_i(x_i^{k-1};S_i^k)$
    $y_i^{k+1}=\sum_{j=1}^m a_{i,j}(k)(y_j^k+v_j^k-v_j^{k-1})$
  $x_i^{k+1}=\sum_{j=1}^m a_{i,j}(k)(x_j^k-\eta y_j^{k+1})$
Return: Choose x_{out} uniformly at random from $\big\{x_i^k\big\}_{k=1,\cdots,K,i=1,\cdots,m}$

Preliminaries for DGFM$^+$. Mini-batch zeroth-order oracle estimator plays a key role in DGFM$^+$. The variance of the gradient estimator can be reduced by increasing the batch size. Furthermore, the smoothness merit of randomized smoothing for mini-batch zeroth-order oracle estimator is still established. All these properties are stated in the following Proposition 4.1.

Proposition 4.1 (Corollary 2.1 and Proposition 2.4 (Chen, Xu, and Luo 2023))**.** *Under Assumption 1, it holds that $\mathbb{E}_S\big[\big\|g_i(x;S)-\nabla f_\delta^i(x)\big\|^2\big] \le \sigma^2/b$, where $\sigma^2 = 16\sqrt{2\pi}dL_f^2$. Furthermore, for any $w\in\mathbb{S}^{d-1}$ and $x,y\in\mathbb{R}^d$, it holds that $\mathbb{E}_\xi\big[\big\|g_i(x;w,\xi)-g_i(y;w,\xi)\big\|^2\big] \le d^2L_f^2\delta^{-2}\|x-y\|^2$.*

Algorithm and Convergence Analysis. In DGFM$^+$, we divide all the K iterations into R cycles, each containing T iterations. In the first iteration of each cycle, we sample a mini-batch of size b' to compute the stochastic gradient, and then use batchsize of b for the rest iterations. In addition, we use the SPIDER (Fang et al. 2018) method to track the gradient for variance reduction. For the decentralized setting, we perform gradient tracking on the obtained gradients to approximate the gradient of the finite sum function and finally perform gradient descent on the variable x. It is worth noting that we need to restart the gradient tracking at the beginning of each cycle. To reduce the consensus error, we perform frequent fast communication $\mathcal{T}$ times. DGFM$^+$ is given in Algorithm 2.

Lemma 4.1. *Let $\{x_i^k,y_i^k,g_i^k,v_i^k\}$ be the sequence generated by DGFM$^+$ and $\{\mathbf{x}^k,\mathbf{y}^k,\mathbf{g}^k,\mathbf{v}^k\}$ be the corresponding stack variables, then for $k\ge 0$, we have*

$$\mathbf{x}^{k+1}=\tilde{A}(k)(\mathbf{x}^k-\eta\mathbf{y}^{k+1}),\bar{\mathbf{x}}^{k+1}=\bar{\mathbf{x}}^k-\eta\bar{\mathbf{y}}^{k+1}$$
$$\text{and}\quad \bar{\mathbf{y}}^{k+1}=\bar{\mathbf{v}}^k.$$

Furthermore, for $rT<k<(r+1)T$, $r=0,\cdots,R-1$, we have

$$\mathbf{y}^{k+1}=\tilde{A}(k)(\mathbf{y}^k+\mathbf{v}^k-\mathbf{v}^{k-1}),$$
$$\bar{\mathbf{y}}^{k+1}=\bar{\mathbf{y}}^k+\bar{\mathbf{v}}^k-\bar{\mathbf{v}}^{k-1}.$$

The purpose of restart gradient tracking is to ensure that $\bar{\mathbf{y}}^{k+1}=\bar{\mathbf{v}}^k$ holds throughout the entire sequence. Additionally, it helps to truncate the accumulation of the variance bound described in Lemma 4.4. However, this operation may introduce a consensus error at the beginning of each cycle. Inspired from (Luo and Ye 2022; Chen, Ye, and Luo 2022), we perform multiple rounds of communication before the start of each cycle. This will be further explained in detail in Lemma 4.2. Next, we give some consensus results for DGFM$^+$.

Lemma 4.2. *For sequence $\{\mathbf{x}^k,\mathbf{y}^k\}$ generated by DGFM$^+$, we have*

$$\begin{aligned}&\mathbb{E}\big[\big\|\mathbf{y}_\perp^{k+1}\big\|^2\big]-\rho^2(1+\alpha_1)\mathbb{E}\big[\big\|\mathbf{y}_\perp^k\big\|^2\big]\\&\le 3\eta^2\Big(\frac{\rho^2L_\delta^2}{\alpha_1}+\frac{\rho^2d^2L_f^2}{b\delta^2}\Big)\mathbb{E}\big[\big\|\bar{\mathbf{v}}^{k-1}\big\|^2\big]\\&\quad+\theta_9\big(\mathbb{E}\big[\big\|\mathbf{x}_\perp^k\big\|^2\big]+\mathbb{E}\big[\big\|\mathbf{x}_\perp^{k-1}\big\|^2\big]\big),\\&\quad rT+1\le k<(r+1)T,\forall r=0,1,\cdots,R-1,\end{aligned}$$

$$\mathbb{E}(\mathbf{y}_\perp^{rT+1})\le 2\rho^{\mathcal{T}}m(\sigma^2+L_f^2),\forall r=0,\cdots,R-1,$$

and for any $k\ge 0$,

$$\mathbb{E}\big[\big\|\mathbf{x}_\perp^{k+1}\big\|^2\big]\le(1+\alpha_2)\rho^2\mathbb{E}\big[\big\|\mathbf{x}_\perp^k\big\|^2\big]+\theta_{10}\mathbb{E}\big[\big\|\mathbf{y}_\perp^{k+1}\big\|^2\big],$$

where $\theta_9=3(\rho^2L_\delta^2/\alpha_1+\rho^2d^2L_f^2/\delta^2)$ and $\theta_{10}=(1+\alpha_2^{-1})\rho^2\eta^2$.

Similar to the proof process of DGFM, we will present the descent properties of the average sequence in Lemma 4.3. Since SPIDER is used here, we also provide an upper bound for variance in Lemma 4.4.

Lemma 4.3. *For the sequence $\{\bar{x}^k, \bar{v}^k\}$ generated by DGFM$^+$ and $f_\delta(x) = \frac{1}{m}\sum_{i=1}^m f_\delta^i(x)$, we have*

$$f_\delta(\bar{x}^{k+1}) - f_\delta(\bar{x}^k) + \frac{\eta}{2}\|\nabla f_\delta(\bar{x}^k)\|^2 \le -\big(\frac{\eta}{2} - \frac{L_\delta \eta^2}{2}\big)\|\bar{v}^k\|^2 + \frac{\eta}{2}\|\nabla f_\delta(\bar{x}^k) - \bar{v}^k\|^2.$$

Lemma 4.4. *Let $\{\bar{x}^k, \bar{v}^k\}$ be the sequence generated by DGFM$^+$, then for $rT \le k' < k \le (r+1)T - 1$, we have*

$$\mathbb{E}\big[\|\bar{v}^k - \nabla f_\delta(\bar{x}^k)\|^2\big] - \frac{6\eta^2 d^2 L_f^2}{m^2\delta^2 b}\sum_{j=k'}^{k}\mathbb{E}\big[\|\bar{\mathbf{v}}^{j-1}\|^2\big]$$

$$\le \frac{2L_\delta^2}{m}\mathbb{E}\big[\|\mathbf{x}_\perp^k\|^2\big] + \frac{6d^2L_f^2}{m^2\delta^2 b}\sum_{j=k'}^{k}\mathbb{E}\big[\|\mathbf{x}_\perp^j\|^2\big] + \mathbb{E}\big[\|\mathbf{x}_\perp^{j-1}\|^2\big] + 2\mathbb{E}\left[\Big\|\bar{v}^{k'} - \frac{1}{m}\sum_{i=1}^m \nabla f_\delta^i(x_i^{k'})\Big\|^2\right].$$

Moreover, for $k = rT$, $r = 0, \cdots, R-1$, we have $\mathbb{E}\big[\|\bar{v}^k - \frac{1}{m}\sum_{i=1}^m \nabla f_\delta^i(x_i^k)\|^2\big] \le \sigma^2/b'$.

By multiplying the consensus error descent of x, y shown in Lemma 4.2 with their corresponding coefficients, β_x and β_y, and combining it with the results of mean sequence (Lemma 4.3) and variance upper bound (Lemma 4.4), we can obtain the following Lemma 4.5.

Lemma 4.5 (Informal)**.** *For the sequence $\{\mathbf{x}^k, \mathbf{y}^k\}$ generated by DGFM$^+$, we have*

$$\frac{\eta}{2}\sum_{k=0}^{RT-1}\mathbb{E}\big[\|\nabla f_\delta(\bar{x}^k)\|^2\big] + \theta_{11}\sum_{k=0}^{RT-1}\mathbb{E}\big[\|\mathbf{x}_\perp^k\|^2\big]$$
$$\le -\theta_{12}\sum_{k=0}^{RT-1}\mathbb{E}\big[\|\bar{v}^k\|^2\big] - \beta_x\mathbb{E}(\mathbf{x}_\perp^{RT+1})$$
$$- \theta_{13}\sum_{k=0}^{RT-1}\mathbb{E}\big[\|\mathbf{y}_\perp^k\|^2\big] - \theta_{14}\mathbb{E}\big[\|\mathbf{y}_\perp^{RT}\|^2\big] - \mathbb{E}\big[f_\delta(\bar{x}^{RT})\big]$$
$$+ \mathbb{E}[f_\delta(\bar{x}^0)] + 2\rho^{\mathcal{T}} m(\sigma^2 + L_f^2)\cdot\beta_y R + \theta_{15},$$

with constants $(\theta_{11}, \theta_{12}, \theta_{13}, \theta_{14}, \theta_{15})$ depend on $(\alpha_1, \alpha_2, b, \beta_x, \beta_y, d, \eta, L_\delta, L_f, R, m, \rho, T)$.

Similar to Lemma 3.4, Lemma 4.5 gives an upper bound of $\mathbb{E}\big[\|\nabla f_\delta(\bar{x}^k)\|^2\big] + (\theta_{11}/\eta)\cdot\mathbb{E}\big[\|\mathbf{x}^k - \bar{\mathbf{x}}^k\|^2\big]$, which can be used to derive the complexity of Goldstein stationary point x_i^k in expectation. We present specific parameter setting in the following Theorem.

Theorem 4.1 (Informal)**.** *DGFM$^+$ can output a (δ, ε)-Goldstein stationary point of $f(\cdot)$ in expectation with total stochastic zeroth-order complexity at most $\mathcal{O}(\max\{\Delta_\delta\delta^{-1}\varepsilon^{-3}d^{\frac{3}{2}}m^{-\frac{3}{2}}, \Delta_\delta\varepsilon^{-\frac{7}{2}}d^{\frac{3}{2}}m^{-\frac{1}{2}}\})$ and the total communication rounds is at most $\mathcal{O}\big((\varepsilon^{-1} + \log(\varepsilon^{-1}d))\cdot\max\{\Delta_\delta\varepsilon^{-\frac{3}{2}}d^{\frac{1}{2}}m^{\frac{1}{2}}, \Delta_\delta\varepsilon^{-2}d^{\frac{1}{2}}m^{-\frac{1}{2}}\}\big)$ by setting $\alpha_1 = \alpha_2 = (1-\rho^2)/2\rho^2$, $\delta = \mathcal{O}(\varepsilon)$, $\beta_x = \mathcal{O}(\delta^{-1})$, $\beta_y = \mathcal{O}(\delta)$, $b' = \mathcal{O}(\varepsilon^{-2}), b = \mathcal{O}(\varepsilon^{-1})$, $\eta = \mathcal{O}(\varepsilon)$, $T = \mathcal{O}(\varepsilon^{-1})$, $R = \mathcal{O}(\Delta_\delta\varepsilon^{-2})$, $\mathcal{T} = \mathcal{O}(\log(\varepsilon^{-1}))$.*

Dataset	n	d	Dataset	n	d
a9a	32,561	123	w8a	49,749	300
HIGGS	11,000,000	28	covtype	581,012	54
rcv	20,242	47,236	SUSY	5,000,000	18
ijcnn1	49,990	22	skin_nonskin	245,057	3

Table 1: Descriptions of datasets used in our experiments.

Remark 2. *Compared to DGFM, DGFM$^+$ requires less evaluations of zeroth-order oracles to achieve the same level of accuracy, which is consistent with the findings of (Chen, Xu, and Luo 2023). Additionally, DGFM$^+$ involves significantly fewer communication rounds than DGFM, and the number of communication rounds required is of the same order as the number of iterations K.*

Remark 3. *If we do not use multiple rounds of communication during the restart of gradient tracking and only perform communication once, i.e., $\mathcal{T} = 1$, we can achieve the same complexity result to DGFM by setting the parameters appropriately, i.e., $\alpha_1 = \alpha_2 = (1-\rho^2)/2\rho^2, \delta = \mathcal{O}(\varepsilon), \beta_x = \mathcal{O}(\varepsilon^{-2}), \beta_y = \mathcal{O}(\varepsilon^2)$, $b' = \mathcal{O}(\varepsilon^{-2})$, $b = \mathcal{O}(1)$, $T = \mathcal{O}(\varepsilon^{-2})$, $R = \mathcal{O}(\varepsilon^{-2})$ and $\eta = \mathcal{O}(\varepsilon^2)$. However, this parameter setting will increase the number of communication rounds to $\mathcal{O}(\delta^{-1}\varepsilon^{-3})$, which is one order of magnitude higher than the current result in Theorem 4.1.*

5 Numerical Study

In this section, we show the outperformance of DGFM and DGFM$^+$ via some numerical experiments.

5.1 Non-convex SVM with Capped-ℓ_1 Penalty

The first experiment considers the model of penalized Support Vector Machines (SVM). We aim to find a hyperplane to separate data points into two categories. To enhance the robustness of the classifier, we introduce the non-convex and non-smooth regularizers.

Data: We evaluate our proposed algorithms using several standard datasets in LIBSVM (Chang and Lin 2011), which are described in Table 1. The feature vectors of all datasets are normalized before optimization.

Model: We consider the non-convex penalized SVM with capped-ℓ_1 regularizer (Zhang 2010). The model aims at training a binary classifier $x \in \mathbb{R}^d$ on the training data $\{a_i, b_i\}_{i=1}^n$, where $a_i \in \mathbb{R}^d$ and $b_i \in \{1, -1\}$ are the feature of the i-th sample and its label, respectively. For DGFM$^+$ and DGFM, the objective function can be written as

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

where $f^i(x) = \frac{1}{n_i}\sum_{j=1}^{n_i}\ell(b_i^j(a_i^j)^\top x) + \gamma(x)$, $\ell(x) = \max\{1-x, 0\}, \gamma(x) = \lambda\sum_{j=1}^d \min\{|x_j|, \alpha\}$, $n = \sum_{i=1}^m n_i$ and $\lambda, \alpha > 0$. Similar to (Chen, Xu, and Luo 2023), we take $\lambda = 10^{-5}/n$ and $\alpha = 2$.

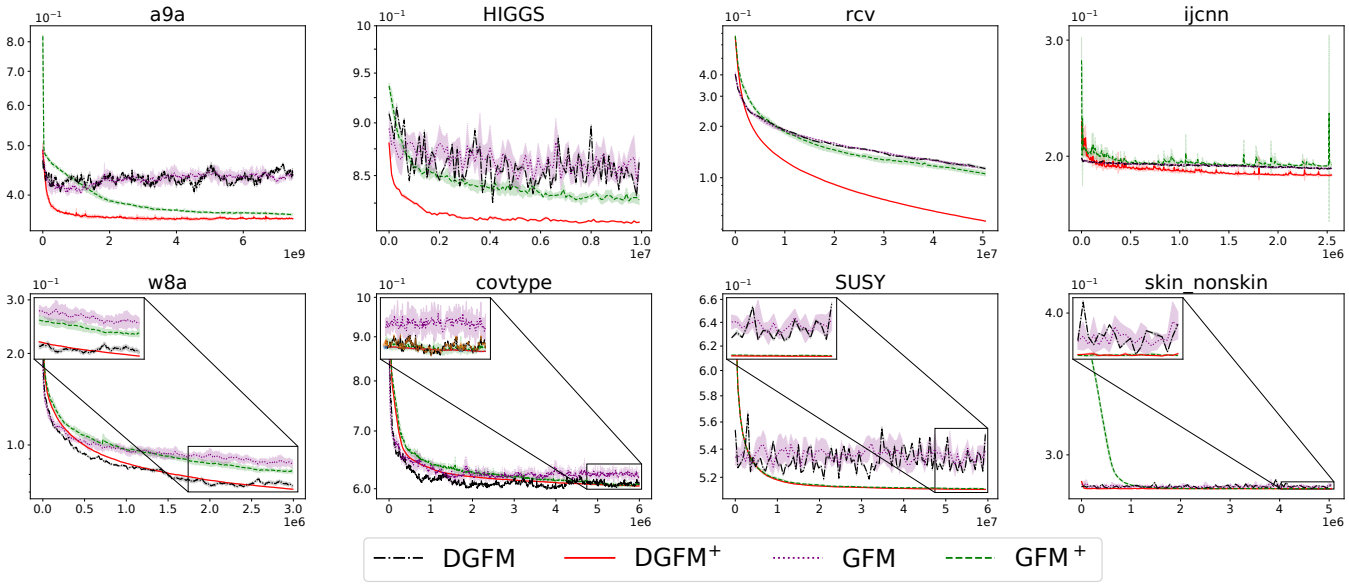

Figure 1: We assess the convergence performance of four algorithms by plotting the objective function value on the y-axis against the number of zeroth-order calls on the x-axis.

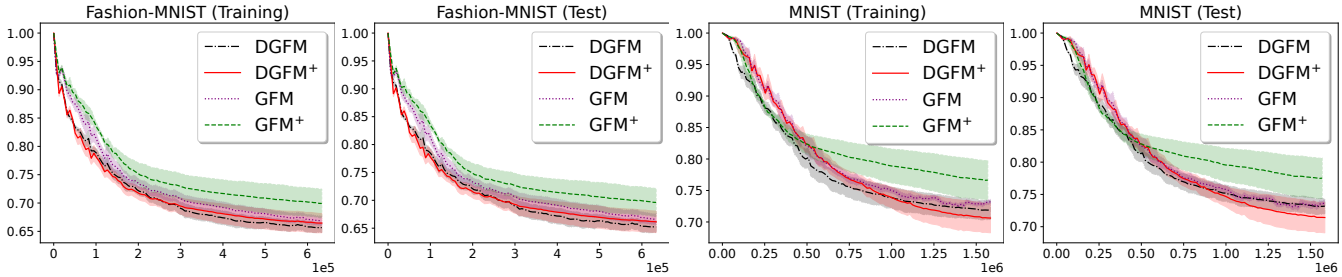

Figure 2: We assess the attacking performance of four algorithms by plotting the accuracy after attacking on the y-axis against the number of zeroth-order calls on the x-axis.

Network topology: We consider a simple ring-based topology of the communication network. We set the number of worker nodes to $m = 20$. The setting can be found in (Xian et al. 2021).

Performance measures: We measure the performance of the decentralized algorithms by the decrease of the global cost function value $f(\bar{x})$, to which we refer as loss versus zeroth-order gradient calls.

Comparison: We compare the proposed DGFM and DGFM$^+$ with GFM (Lin, Zheng, and Jordan 2022) and GFM$^+$ (Chen, Xu, and Luo 2023). Throughout all the experiments, we set $\delta = 0.001$ and tune the stepsize η from $\{0.0005, 0.001, 0.005, 0.01\}$ for all four algorithms and b' from $\{10, 100, 500\}$, T from $\{10, 50, 100\}$ for DGFM$^+$ and GFM$^+$, $\mathcal{T}$ from $\{1, 5, 10\}$ for DGFM$^+$, $m = 20$ for two decentralized algorithms. We run all the algorithms with the same number of calls to the zeroth-order oracles. We plot the average of five runs in Figure 1.

In most of the experiments we conducted, our decentralized algorithms significantly outperform their serial counterparts. While DGFM sometimes exhibits slow convergence due to high consensus error, DGFM$^+$ consistently demonstrates more robust performance and often achieves the best results. In larger sample size test cases (SUSY, HIGGS), DGFM$^+$ often demonstrates notably faster convergence than DGFM. These outcomes further corroborate our theoretical analysis of DGFM and DGFM$^+$. Additionally, we note that both DGFM$^+$ and GFM$^+$ have exhibited significantly stable performance, as seen in their smoother convergence curves, while DGFM and GFM show more fluctuation. This observation aligns with the theoretical advantage offered by the variance reduction technique.

5.2 Universal Attack

We consider the black-box adversarial attack on image classification with LeNet (LeCun et al. 1998). Our objective is to discover a universal adversarial perturbation (Moosavi-Dezfooli et al. 2017). When applied to the original image, this perturbation induces misclassification in machine learning models while remaining inconspicuous to human observers. Network topology and Performance measures are the same as in the previous experiment. However, we set $m = 8$ for two decentralized algorithms here.

Dataset	Fashion-MNIST		MNIST	
	Training	Test	Training	Test
DGFM	**65.66**(**0.88**)	**65.22**(1.07)	71.92(1.56)	73.18(1.21)
DGFM$^+$	66.46(1.80)	66.14(2.03)	**70.62**(2.47)	**71.42**(2.50)
GFM	66.86(0.90)	66.64(**1.02**)	73.20(**0.41**)	73.66(**0.55**)
GFM$^+$	69.92(2.67)	69.62(2.63)	76.64(3.29)	77.52(3.29)

Table 2: Attacking result (Accuracy/std%)

Data: We evaluate our proposed algorithms using two standard datasets, MNIST and Fashion-MNIST.

Model: The problem can be formulated as the following non-smooth non-convex problem:

$$\min_{\|\zeta\|_\infty \le \kappa} \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{|\mathcal{D}_i|}\sum_{j=1}^{|\mathcal{D}_i|} -\ell(a_i^j+\zeta, b_i^j)\right),$$

where the dataset $\mathcal{D}_i = \{a_i^j, b_i^j\}$, a_i^j represents the image features, $b_i^j \in \mathbb{R}^C$ is the one-hot encoding label, C is the number of classes, $|\mathcal{D}_i|$ is the cardinality of dataset $\mathcal{D}_i$, κ is the constraint level of the distortion, ζ is the perturbation vector and $\ell(\cdot,\cdot)$ is the cross entropy function. We set $\kappa = 0.25$ for Fashion-MNIST and $\kappa = 0.5$ for MNIST. Following the setup in the work of (Chen, Xu, and Luo 2023), we iteratively perform an additional projection step for constraint satisfaction.

Comparison: We use two pre-trained models with 99.29% accuracy on MNIST and 92.30% accuracy on Fashion-MNIST, respectively. We use GFM, GFM$^+$, DGFM and DGFM$^+$ to attack the pre-trained LeNet on 59577 images of MNIST and 55384 images of Fashion-MNIST, which are classified correctly on the train set for training LeNet. Furthermore, we evaluate the perturbation on a dataset comprising 9885 MNIST images and 8975 Fashion-MNIST images. These images have been accurately classified during the testing phase of the LetNet training. Throughout all the experiments, we set $\delta = 0.01$, $b = \{16, 32, 64\}$. For DGFM$^+$ and GFM$^+$, we tune b' from $\{40, 80, 800, 1600\}$, T from $\{2, 5, 10, 20\}$. Additionally, tune $\mathcal{T}$ from $\{1, 10, 20\}$ for DGFM$^+$. For all algorithms, we tune the stepsize η from $\{0.05, 0.1, 0.5, 1\}$ and multiply a decay factor 0.6 if no improvement in 300 iterations. For all experiments, we set the initial perturbation as $\mathbf{0}$. The results of the average of five runs are shown in Figure 2 and Table 2. On the Fashion-MNIST dataset, DGFM achieves the lowest accuracy after attacking and small variance. For MNIST, DGFM$^+$ achieves the lowest accuracy after attacking, but its variance is larger. In general, we can observe that our algorithms perform better than the serial counterparts.

Acknowledgments

Qi Deng is partially supported by National Natural Science Foundation of China(NSFC) [Grant NSFC-11831002, 72150001]; Luo Luo is supported by National Natural Science Foundation of China (No. 62206058) and Shanghai Sailing Program (22YF1402900).

References

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.

Chen, J.; Jordan, M. I.; and Wainwright, M. J. 2020. HopSkipJumpAttack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*, 1277–1294. IEEE.

Chen, L.; Xu, J.; and Luo, L. 2023. Faster Gradient-Free Algorithms for Nonsmooth Nonconvex Stochastic Optimization. In *International Conference on Machine Learning*, 5219–5233. PMLR.

Chen, L.; Ye, H.; and Luo, L. 2022. An Efficient Stochastic Algorithm for Decentralized Nonconvex-Strongly-Concave Minimax Optimization. *arXiv preprint arXiv:2212.02387*.

Chen, S.; Garcia, A.; and Shahrampour, S. 2021. On distributed nonconvex optimization: Projected subgradient method for weakly convex problems in networks. *IEEE Transactions on Automatic Control*, 67(2): 662–675.

Clarke, F. H. 1990. *Optimization and nonsmooth analysis*. SIAM.

Cutkosky, A.; Mehta, H.; and Orabona, F. 2023. Optimal Stochastic Non-smooth Non-convex Optimization through Online-to-Non-convex Conversion. *arXiv preprint arXiv:2302.03775*.

Davis, D.; Drusvyatskiy, D.; Lee, Y. T.; Padmanabhan, S.; and Ye, G. 2022. A gradient sampling method with complexity guarantees for lipschitz functions in high and low dimensions. *Advances in Neural Information Processing Systems*, 35: 6692–6703.

Di Lorenzo, P.; and Scutari, G. 2016. Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2): 120–136.

Fang, C.; Li, C. J.; Lin, Z.; and Zhang, T. 2018. SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31.

Giannakis, G. B.; Ling, Q.; Mateos, G.; Schizas, I. D.; and Zhu, H. 2017. Decentralized learning for wireless communications and networking. In *Splitting Methods in Communication, Imaging, Science, and Engineering*, 461–497. Springer.

Goldstein, A. 1977. Optimization of Lipschitz continuous functions. *Mathematical Programming*, 13: 14–22.

Hajinezhad, D.; Hong, M.; and Garcia, A. 2019. ZONE: Zeroth-order nonconvex multiagent optimization over networks. *IEEE Transactions on Automatic Control*, 64(10): 3995–4010.

Hong, M.; Hajinezhad, D.; and Zhao, M.-M. 2017. Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks. In *International Conference on Machine Learning*, 1529–1538. PMLR.

Kornowski, G.; and Shamir, O. 2021. Oracle complexity in nonsmooth nonconvex optimization. *Advances in Neural Information Processing Systems*, 34: 324–334.

Kornowski, G.; and Shamir, O. 2023. An Algorithm with Optimal Dimension-Dependence for Zero-Order Nonsmooth Nonconvex Stochastic Optimization. *arXiv preprint arXiv:2307.04504*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lin, T.; Zheng, Z.; and Jordan, M. I. 2022. Gradient-free methods for deterministic and stochastic nonsmooth nonconvex optimization. *Advances in Neural Information Processing Systems*, 35: 26160–26175.

Ling, Q.; and Tian, Z. 2010. Decentralized sparse signal recovery for compressive sleeping wireless sensor networks. *IEEE Transactions on Signal Processing*, 58(7): 3816–3827.

Liu, S.; Chen, J.; Chen, P.-Y.; and Hero, A. 2018. Zeroth-order online alternating direction method of multipliers: Convergence analysis and applications. In *International Conference on Artificial Intelligence and Statistics*, 288–297. PMLR.

Liu, S.; Chen, P.-Y.; Kailkhura, B.; Zhang, G.; Hero III, A. O.; and Varshney, P. K. 2020. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5): 43–54.

Lu, S.; Zhang, X.; Sun, H.; and Hong, M. 2019. GNSD: A gradient-tracking based nonconvex stochastic algorithm for decentralized optimization. In *2019 IEEE Data Science Workshop (DSW)*, 315–321. IEEE.

Luo, L.; and Ye, H. 2022. An Optimal Stochastic Algorithm for Decentralized Nonconvex Finite-sum Optimization. *arXiv preprint arXiv:2210.13931*.

Mancino-Ball, G.; Miao, S.; Xu, Y.; and Chen, J. 2023. Proximal stochastic recursive momentum methods for nonconvex composite decentralized optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 9055–9063.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1765–1773.

Nedic, A.; Olshevsky, A.; and Shi, W. 2017. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4): 2597–2633.

Nesterov, Y.; and Spokoiny, V. 2017. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17: 527–566.

Pan, T.; Liu, J.; and Wang, J. 2020. D-SPIDER-SFO: A decentralized optimization algorithm with faster convergence rate for nonconvex problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1619–1626.

Sahu, A. K.; Jakovetic, D.; Bajovic, D.; and Kar, S. 2018. Distributed zeroth order optimization over random networks: A Kiefer-Wolfowitz stochastic approximation approach. In *2018 IEEE Conference on Decision and Control (CDC)*, 4951–4958. IEEE.

Scutari, G.; and Sun, Y. 2019. Distributed nonconvex constrained optimization over time-varying digraphs. *Mathematical Programming*, 176: 497–544.

Shamir, O. 2017. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *Journal of Machine Learning Research*, 18(1): 1703–1713.

Sun, H.; and Hong, M. 2019. Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms. *IEEE Transactions on Signal processing*, 67(22): 5912–5928.

Sun, H.; Lu, S.; and Hong, M. 2020. Improving the sample and communication complexity for decentralized nonconvex optimization: Joint gradient estimation and tracking. In *International conference on machine learning*, 9217–9228. PMLR.

Sun, Y.; Scutari, G.; and Daneshmand, A. 2022. Distributed optimization based on gradient tracking revisited: Enhancing convergence rate via surrogation. *SIAM Journal on Optimization*, 32(2): 354–385.

Tang, Y.; Ren, Z.; and Li, N. 2023. Zeroth-order feedback optimization for cooperative multi-agent systems. *Automatica*, 148: 110741.

Tang, Y.; Zhang, J.; and Li, N. 2020. Distributed zero-order algorithms for nonconvex multiagent optimization. *IEEE Transactions on Control of Network Systems*, 8(1): 269–281.

Tian, L.; Zhou, K.; and So, A. M.-C. 2022. On the finite-time complexity and practical computation of approximate stationarity concepts of lipschitz functions. In *International Conference on Machine Learning*, 21360–21379. PMLR.

Tsitsiklis, J. N. 1984. Problems in decentralized decision making and computation. Technical report, Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems.

Vogels, T.; He, L.; Koloskova, A.; Karimireddy, S. P.; Lin, T.; Stich, S. U.; and Jaggi, M. 2021. Relaysum for decentralized deep learning on heterogeneous data. *Advances in Neural Information Processing Systems*, 34: 28004–28015.

Wang, J.; Hu, J.; Chen, S.; Deng, Z.; and So, A. M.-C. 2023. Decentralized Weakly Convex Optimization Over the Stiefel Manifold. *arXiv preprint arXiv:2303.17779*.

Wang, Z.; Zhang, J.; Chang, T.-H.; Li, J.; and Luo, Z.-Q. 2021. Distributed stochastic consensus optimization with momentum for nonconvex nonsmooth problems. *IEEE Transactions on Signal Processing*, 69: 4486–4501.

Xian, W.; Huang, F.; Zhang, Y.; and Huang, H. 2021. A faster decentralized algorithm for nonconvex minimax problems. *Advances in Neural Information Processing Systems*, 34: 25865–25877.

Xiao, L.; Boyd, S.; and Lall, S. 2005. A scheme for robust distributed sensor fusion based on average consensus. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, 63–70. IEEE.

Xiao, T.; Chen, X.; Balasubramanian, K.; and Ghadimi, S. 2023. A One-Sample Decentralized Proximal Algorithm

for Non-Convex Stochastic Composite Optimization. *arXiv preprint arXiv:2302.09766*.

Xin, R.; Das, S.; Khan, U. A.; and Kar, S. 2021. A stochastic proximal gradient framework for decentralized non-convex composite optimization: Topology-independent sample complexity and communication efficiency. *arXiv preprint arXiv:2110.01594*.

Xin, R.; Khan, U.; and Kar, S. 2021a. A hybrid variance-reduced method for decentralized stochastic non-convex optimization. In *International Conference on Machine Learning*, 11459–11469. PMLR.

Xin, R.; Khan, U. A.; and Kar, S. 2021b. An improved convergence analysis for decentralized online stochastic non-convex optimization. *IEEE Transactions on Signal Processing*, 69: 1842–1858.

Xin, R.; Khan, U. A.; and Kar, S. 2022. Fast decentralized nonconvex finite-sum optimization with recursive variance reduction. *SIAM Journal on Optimization*, 32(1): 1–28.

Yu, Z.; Ho, D. W.; and Yuan, D. 2021. Distributed randomized gradient-free mirror descent algorithm for constrained optimization. *IEEE Transactions on Automatic Control*, 67(2): 957–964.

Zeng, J.; and Yin, W. 2018. On nonconvex decentralized gradient descent. *IEEE Transactions on signal processing*, 66(11): 2834–2848.

Zhang, J.; Lin, H.; Jegelka, S.; Sra, S.; and Jadbabaie, A. 2020. Complexity of finding stationary points of nonconvex nonsmooth functions. In *International Conference on Machine Learning*, 11173–11182. PMLR.

Zhang, T. 2010. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11(3).