Accelerated Gradient-Free Decentralized Stochastic Optimization

Haoyuan Cai†, Jie Chen*, and Ali H.Sayed†
† École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
* Northwestern Polytechnical University, Xi'an, China

***Abstract*—Gradient-free (zeroth-order) optimization is well-suited for black-box optimization and memory-efficient learning when gradient information is unavailable or costly to obtain. In this paper, we study decentralized stochastic and finite-sum nonconvex optimization using only function information from a stochastic zeroth-order oracle. We develop coordinate-wise zeroth-order algorithms built on a first-order decentralized learning framework that subsumes a broad class of decentralized strategies, including adapt-then-combine gradient-tracking, exact diffusion, and EXTRA. To further improve the zeroth-order gradient estimator, we incorporate a probabilistic variance reduction strategy based on the GRACE acceleration strategy. We establish convergence guarantees for the proposed methods over sparsely connected networks and identify parameter regimes that achieve linear speedup with respect to the number of agents K in stochastic settings. In particular, the zeroth-order exact diffusion method attains a favorable per-agent function query complexity of $\mathcal{O}\Big(\frac{d\varepsilon^{-3}}{K}+\frac{d\varepsilon^{-2}}{(1-\lambda)^2}\Big)$ in the stochastic regime, and $\mathcal{O}\Big(\frac{d\sqrt{N}\varepsilon^{-2}}{\sqrt{K}(1-\lambda)^2}\Big)$ in the finite-sum regime, among others, for finding a ε-stationary point, where λ denotes the second largest eigenvalue in magnitude of the communication matrix, N is the local sample size, and d is the problem dimension. Numerical experiments illustrate the effectiveness of the proposed decentralized gradient-free method.**

I. Introduction

Stochastic nonconvex optimization is ubiquitous in modern machine learning and information theoretic applications, including adversarial attacks [1], online sensor networks [2], and privacy-preserving learning [3]. As the scale of learning models continues to grow [4], distributed optimization that leverages multiple machines has become a powerful paradigm for overcoming the computational bottlenecks of single-machine learning. Meanwhile, zeroth-order (ZO) optimization offers a memory-efficient alternative to first-order (FO) methods for large scale training, as it relies solely on forward function evaluations [5]. Motivated by the need for distributed gradient-free optimization, we study decentralized nonconvex optimization problems involving K agents.

$$\min_{w\in\mathbb{R}^d} J(w)=\frac{1}{K}\sum_{k=1}^{K}J_k(w), \tag{1}$$

$$J_k(w)=\begin{cases}\mathbb{E}_{\boldsymbol{\zeta}_k}[Q_k(w;\boldsymbol{\zeta}_k)] & \textbf{(Stochastic)}\\ \frac{1}{N}\sum_{s=1}^{N}Q_k(w;\boldsymbol{\zeta}_k(s)) & \textbf{(Finite-sum)}\end{cases}$$

where the local cost $J_k(w)$ is assumed to be L-smooth and nonconvex over $w\in\mathbb{R}^d$, while it is either defined as the expectation of a loss $Q_k(w;\boldsymbol{\zeta}_k)$ in the *stochastic* setting, or as a sum of local losses $\{Q_k(w;\boldsymbol{\zeta}_k(s))\}$ in the *finite-sum* empirical setting, where $\boldsymbol{\zeta}_k(s)$ denotes the sth sample from the ordered local dataset $\{\boldsymbol{\zeta}_k(s)\}_{s=1}^N$. We focus on the gradient-free setting where an agent is assumed to have access only to stochastic ZO oracle evaluations of the loss value $Q_k(w;\boldsymbol{\xi}_k)$, where $\boldsymbol{\xi}_k=\boldsymbol{\zeta}_k$ (stochastic) or $\boldsymbol{\xi}_k=\boldsymbol{\zeta}_k(s)$ (finite-sum).

A substantial body of work has studied ZO stochastic optimization in a single-machine setting [6]–[9]. However, these methods cannot be directly used in decentralized environments. To address the limitation, a growing line of works proposed decentralized ZO optimization algorithms [10]–[15]. Early work in [10] proposed an adapt-then-combine (ATC) ZO diffusion learning algorithm. Subsequently, the work [11] studied a variance-reduction ZO diffusion strategy based on the randomized ZO technique. The work in [12] investigated a gradient tracking-based ZO algorithm, while [13] developed a decentralized primal-dual ZO method using a randomized ZO strategy. More recently, [14] analyzed a single-point-based ZO decentralized optimization scheme, and [15] proposed a momentum-based ZO method. In contrast to these prior works, this paper investigates a unified decentralized ZO framework, inspired by the FO decentralized framework SUDA [16]–[18]. Rather than directly replacing FO stochastic gradients with ZO estimates within SUDA, we integrate coordinate-wise ZO gradient estimation with a probabilistic variance reduction strategy based on the GRACE estimator [19], [20]. As a result, the proposed approach effectively handles both stochastic and finite-sum optimization problems in a fully decentralized gradient-free setting.

Contributions. (i) We develop decentralized coordinate-wise ZO algorithms built on the unified FO decentralized framework SUDA [16]–[18], which operates in a gradient-free environment and remains robust to data heterogeneity challenge; (ii) we enhance the coordinate-wise ZO gradient estimator by incorporating the recently developed accelerated GRACE estimator [19], [20], leading to improved convergence speed and achieving linear speedup with respect to the number of agents K in stochastic settings; (iii) we establish unified performance bounds for a broad class of decentralized ZO strategies and further specialize the results to both stochastic and finite-sum optimization settings.

II. Algorithm Development

Notation. We use regular font (e.g., w) to denote deterministic quantities, and bold font (e.g., $\boldsymbol{w}$) to denote stochastic quantities. We use a calligraphic font, e.g., $\boldsymbol{\mathcal{x}}=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_K\}$, to

denote the concatenated variables. The notation $\otimes$ stands for the Kronecker product, $\mathbf{I}_K$ denotes the $K \times K$ identity matrix, and $\mathbb{1}_K$ denotes the K-dimensional one vector.

A. A brief review of unified framework

Recent works [16]–[18] introduced SUDA (**s**tochastic **u**nified **d**ecentralized **a**lgorithm) for decentralized nonconvex optimization, demonstrating a unified framework that brings together several existing decentralized strategies within a single formulation. This approach is primarily designed for stochastic FO optimization, and a straightforward extension to a gradient-free environment can be inefficient. We address this issue by integrating acceleration techniques. To this end, we first review the mathematical formulation of SUDA below.

Consider the block network model and the stochastic gradient at iteration i

$$\boldsymbol{\mathcal{W}}_i \triangleq \operatorname{col}\{\boldsymbol{w}_{1,i}, \ldots, \boldsymbol{w}_{K,i}\}, \tag{2}$$

$$\nabla \mathcal{Q}(\boldsymbol{\mathcal{W}}_i; \boldsymbol{\xi}_i) \triangleq \operatorname{col}\{\nabla Q_1(\boldsymbol{w}_{1,i}; \boldsymbol{\xi}_{1,i}), \ldots, \nabla Q_K(\boldsymbol{w}_{K,i}; \boldsymbol{\xi}_{K,i})\}, \tag{3}$$

SUDA recursively performs the following steps for all $i \geq 0$:

$$\boldsymbol{\mathcal{W}}_{i+1} = \mathcal{A}(\mathcal{C}\,\boldsymbol{\mathcal{W}}_i - \mu \nabla \mathcal{Q}(\boldsymbol{\mathcal{W}}_i; \boldsymbol{\xi}_i)) - \mathcal{B}\,\boldsymbol{\mathcal{D}}_i, \tag{4}$$

$$\boldsymbol{\mathcal{D}}_{i+1} = \boldsymbol{\mathcal{D}}_i + \mathcal{B}\,\boldsymbol{\mathcal{W}}_{i+1}, \tag{5}$$

where $\mu > 0$ is the learning rate, $\boldsymbol{\mathcal{D}}_i \in \mathbb{R}^{Kd \times Kd}$ is the dual variable, and $\{\mathcal{A}, \mathcal{B}, \mathcal{C}\} \in \mathbb{R}^{Kd \times Kd}$ are design matrices. We note that the unified forms in (4) and (5) are intended solely as compact representations for theoretical analysis and are not meant to be implemented directly. To implement the algorithm efficiently, one can choose the design matrices $\{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$, for instance, as $\{W \otimes \mathbf{I}_d, (\mathbf{I}_{Kd} - W \otimes \mathbf{I}_d)^{\frac{1}{2}}, \mathbf{I}_{Kd}\}$, where $W = [W_{k\ell}] \in \mathbb{R}^{K \times K}$ is the communication matrix. Under this choice, the SUDA recursion reduces to the exact diffusion (ED) algorithm, in which each agent recursively performs the following updates for all $i \geq 0$:

$$\boldsymbol{w}_{k,i+1} = \sum_{\ell \in \mathcal{N}_k} W_{k\ell} \Big[2\boldsymbol{w}_{\ell,i} - \boldsymbol{w}_{\ell,i-1} - \mu \Big(\nabla Q_\ell(\boldsymbol{w}_{\ell,i}; \boldsymbol{\xi}_{\ell,i}) - \nabla Q_\ell(\boldsymbol{w}_{\ell,i-1}; \boldsymbol{\xi}_{\ell,i-1}) \Big) \Big]. \tag{6}$$

We can readily see that the above form is more efficient compared to (4), (5) as there is no need to store the design matrices. We also note that SUDA can be interpreted as a class of primal-dual algorithms derived from solving a constrained consensus optimization problem; refer to references [17], [18].

B. ZO coordinate-wise strategy

When the FO gradient information $\nabla \mathcal{Q}(\boldsymbol{\mathcal{W}}_i; \boldsymbol{\xi}_i)$ is unavailable, it can be approximated using finite differences of perturbed function values [8], leading to ZO methods. Common ZO strategies include randomized Gaussian-smoothing, uniform-smoothing, and coordinate-wise schemes [2], [8]. All these techniques are based on estimating gradients from function value differences, which differ in the construction of perturbation vectors. In this work, we focus on coordinate-wise ZO estimation, which yields more accurate gradient approximations than randomized ZO strategies at the cost of increased function query complexity. That being said, our framework remains promising for randomized strategies. Due to page limits, we leave the extension to randomized strategies as future work. For the coordinate-wise ZO strategy, we query $2d$ perturbed function values $\nabla Q_k(w \pm \delta e_j; \boldsymbol{\xi}_k)$, for each k and all $j \in [d]$, where $\delta > 0$ is the smoothing parameter and e_j denotes the j-th standard basis vector in $\mathbb{R}^d$. The resulting ZO estimator at a point w is then constructed using central differences, given by

$$\widehat{\nabla Q}_k(w; \boldsymbol{\xi}_k) \triangleq \sum_{j=1}^{d} \frac{Q_k(w + \delta e_j; \boldsymbol{\xi}_k) - Q_k(w - \delta e_j; \boldsymbol{\xi}_k)}{2\delta} e_j.$$

Here, δ controls the approximation accuracy of the ZO gradient estimator. According to [7], [21], when the local loss gradient $\nabla Q_k(\cdot;\cdot)$ is L-smooth, the approximation error of $\widehat{\nabla Q}_k(w; \boldsymbol{\xi}_k)$ satisfies $\mathbb{E}\|\widehat{\nabla Q}_k(w; \boldsymbol{\xi}_k) - \nabla Q_k(w; \boldsymbol{\xi}_k)\|^2 \leq \mathcal{O}(dL^2\delta^2)$. Intuitively, a smaller value of δ leads to a more accurate gradient approximation. However, choosing δ too small may result in numerical instability as modern computer systems operate with finite numerical precision. Therefore, δ needs to be carefully selected to balance approximation accuracy and numerical stability. By contrast, under the same smoothing parameter δ, the estimation error of randomized ZO methods scales as $\mathcal{O}(d^2L^2\delta^2)$, which is worse by a factor of $\mathcal{O}(d)$ compared to coordinate-wise construction, but offers greater flexibility in function query complexity.

C. ZO probabilistic variance-reduced estimator

The coordinate-wise ZO gradient $\widehat{\nabla Q}_k(w; \boldsymbol{\xi}_k)$ can be viewed as an estimator of the stochastic FO gradient $\nabla Q_k(w; \boldsymbol{\xi}_k)$. However, because $\nabla Q_k(w; \boldsymbol{\xi}_k)$ is itself a noisy surrogate for the risk gradient $\nabla J_k(w)$, the resulting approximation error of $\widehat{\nabla Q}_k(w; \boldsymbol{\xi}_k)$ is further aggravated, particularly in high-dimensional regimes. To mitigate this issue, we incorporate the coordinate-wise ZO gradient into the GRACE acceleration technique developed in the recent works [19], [20], which is designed to reduce gradient noise in both stochastic and finite-sum optimization problems. GRACE is a probabilistic variance reduced gradient estimator that alternates between two gradient estimators, one of which provides higher-quality gradient information. Moreover, GRACE unifies the momentum-based method STORM [22] and the loopless variance reduction method PAGE [23] as special cases. We describe how GRACE can be applied to accelerate the coordinate-wise ZO gradient. For each agent k, we first generate a Bernoulli random variable $\boldsymbol{\pi}_i \sim \text{Bernoulli}(p)$ using a shared random seed, where p denotes the probability parameter. We then compute

$$\boldsymbol{g}_{k,i} = \begin{cases} \frac{1}{B} \sum_{j=1}^{B} \widehat{\nabla Q}_k(\boldsymbol{w}_{k,i}; \boldsymbol{\xi}_{k,i}^j) & \text{if}(\boldsymbol{\pi}_i = 1) \\ (1-\beta)\Big(\boldsymbol{g}_{k,i-1} - \frac{1}{b} \sum_{j=1}^{b} \widehat{\nabla Q}_k(\boldsymbol{w}_{k,i-1}; \boldsymbol{\xi}_{k,i}^j)\Big) & \\ \quad + \frac{1}{b} \sum_{j=1}^{b} \widehat{\nabla Q}_k(\boldsymbol{w}_{k,i}; \boldsymbol{\xi}_{k,i}^j) & \text{if}(\boldsymbol{\pi}_i = 0). \end{cases} \tag{7}$$

Here, β denotes the smoothing factor. Under the finite-sum setting, the random samples $\{\boldsymbol{\xi}_{k,i}^j\}$ are drawn uniformly without replacement when the event $\boldsymbol{\pi}_i = 1$; otherwise, they are sampled in an independent and identically distributed (i.i.d.) manner. The batch size used when $\boldsymbol{\pi}_i = 1$ is denoted by B, while b denotes the minibatch size when $\boldsymbol{\pi}_i = 0$. Typically, we choose $b \ll B$ to improve computational efficiency, since the update associated with the event $\boldsymbol{\pi}_i = 0$ occurs with higher probability when setting p small. In particular, setting $p = 0$ eliminates the large batch computation step, which favors memory efficiency at the expense of convergence speed. With the proposed probabilistic ZO gradient estimator in (7), we can seamlessly integrate it into the SUDA in (4)-(5).

D. Proposed ZO unified decentralized learning algorithm

As we have previously discussed, the formulation of SUDA in (4)-(5) is primarily introduced for analytical convenience and is not necessarily suited for efficient implementation. In what follows, we present efficient implementations obtained by specifying the combination matrices $\{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$. We focus on three representative choices: ED, ATC-GT, and EXTRA.

Algorithm 1: **z**eroth-**o**rder **e**xact **d**iffusion/**EXTRA** algorithm (**ZO-ED/EXTRA**)

Require: initializing $\boldsymbol{w}_{1,0} = \ldots = \boldsymbol{w}_{K,0} \in \mathbb{R}^d$, $\boldsymbol{g}_{k,-1} = 0\ \forall k \in [K]$, learning rate μ, smoothing factors δ and β, and p, b, B, b_0

for $i = 0$ to $T-1$ **do**
  for each agent k in parallel **do**
    Construct ZO gradient estimator
    if $i == 0$ **then**
      Collect i.i.d. samples $\{\boldsymbol{\xi}_{k,0}^j\}_{j=0}^{b_0}$ and compute $\boldsymbol{g}_{k,0} = \frac{1}{b_0}\sum_{j=1}^{b_0} \widehat{\nabla Q}_k(\boldsymbol{w}_{k,0}; \boldsymbol{\xi}_{k,0}^j)$.
    else
      Update $\boldsymbol{g}_{k,i}$ according to ZO estimator (7).
    end if
    Decentralized learning
    if **ED**:

$$\boldsymbol{w}_{k,i+1} = \sum_{\ell\in\mathcal{N}_k} W_{k\ell}\Big[2\boldsymbol{w}_{\ell,i} - \boldsymbol{w}_{\ell,i-1} - \mu(\boldsymbol{g}_{\ell,i} - \boldsymbol{g}_{\ell,i-1})\Big].$$

    if **EXTRA**:

$$\boldsymbol{w}_{k,i+1} = \sum_{\ell\in\mathcal{N}_k} W_{k\ell}[2\boldsymbol{w}_{\ell,i} - \boldsymbol{w}_{\ell,i-1}] - \mu(\boldsymbol{g}_{\ell,i} - \boldsymbol{g}_{\ell,i-1}).$$

  end for
end for

• **ED**. We choose $\{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$ as $\{W \otimes \mathbf{I}_d, (\mathbf{I}_{Kd} - W \otimes \mathbf{I}_d)^{\frac{1}{2}}, \mathbf{I}_{Kd}\}$ and obtain the **ZO-ED** strategy presented in **Algorithm** 1.

• **EXTRA**. We choose $\{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$ as $\{\mathbf{I}_{Kd}, (\mathbf{I}_{Kd} - W \otimes \mathbf{I}_d)^{\frac{1}{2}}, W \otimes \mathbf{I}_d\}$ and obtain the **ZO-EXTRA** strategy presented in **Algorithm** 1.

• **ZO-ATC-GT**. We choose $\{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$ as $\{W^2 \otimes \mathbf{I}_d, (\mathbf{I}_{Kd} - W \otimes \mathbf{I}_d), \mathbf{I}_{Kd}\}$ and obtain the **ZO-ATC-GT** strategy presented in **Algorithm** 2.

All algorithms initialize identical local variables and appropriate hyperparameters and then iteratively update ZO gradient estimators and local models through decentralized learning.

Algorithm 2 : **z**eroth-**o**rder **a**dapt-**t**hen-**c**ombine **g**radient-**t**racking (**ZO-ATC-GT**)

Require: initializing $\boldsymbol{w}_{1,0} = \ldots = \boldsymbol{w}_{K,0} \in \mathbb{R}^d$, $\boldsymbol{m}_{k,-1} = \boldsymbol{g}_{k,-1} = 0\ \forall k \in [K]$, learning rate μ, smoothing factors δ and β, and p, b, B, b_0

for $i = 0$ to $T-1$ **do**
  for each agent k in parallel **do**
    Construct ZO gradient estimator
    if $i == 0$ **then**
      Collect i.i.d. samples $\{\boldsymbol{\xi}_{k,0}^j\}_{j=0}^{b_0}$ and compute $\boldsymbol{g}_{k,0} = \frac{1}{b_0}\sum_{j=1}^{b_0} \widehat{\nabla Q}_k(\boldsymbol{w}_{k,0}; \boldsymbol{\xi}_{k,0}^j)$.
    else
      Update $\boldsymbol{g}_{k,i}$ according to the ZO estimator (7).
    end if
    Gradient-tracking

$$\boldsymbol{m}_{k,i} = \sum_{\ell\in\mathcal{N}_k} W_{k\ell}[\boldsymbol{m}_{\ell,i-1} - \boldsymbol{g}_{\ell,i-1} + \boldsymbol{g}_{\ell,i}].$$

    Adapt-then-combine

$$\boldsymbol{w}_{k,i+1} = \sum_{\ell\in\mathcal{N}_k} W_{k\ell}[\boldsymbol{w}_{\ell,i} - \mu\boldsymbol{m}_{\ell,i}].$$

  end for
end for

III. Theoretical Analysis

We establish the performance guarantee for the proposed ZO methods and characterize their complexity results in finding an ε-stationary point, i.e., $\mathbb{E}\|\nabla J(\boldsymbol{w}_{c,i})\|^2 \le \varepsilon^2$, where $\boldsymbol{w}_{c,i} \triangleq \frac{1}{K}\sum_{k=1}^K \boldsymbol{w}_{k,i}$ is the network centroid. We first introduce some standard assumptions.

A. Assumptions

Assumption 1. *The global cost is lower bounded, i.e., $J^\star = \inf_w J(w) > -\infty$.*

Assumption 2 (Expected smooth)**.** *Each local loss function $Q_k(w;\boldsymbol{\xi})$ is assumed to be continuously differentiable with an expected L-Lipschitz continuous gradient, i.e.,*

$$\mathbb{E}\|\nabla Q_k(w_1;\boldsymbol{\xi}_k) - \nabla Q_k(w_2;\boldsymbol{\xi}_k)\|^2 \le L^2\|w_1 - w_2\|^2. \quad (8)$$

The above assumption is standard in the variance-reduced ZO literature [7], [24].

Assumption 3 (Bounded variance)**.** *Each local loss function $Q_k(w;\boldsymbol{\xi})$ is assumed to be unbiased and has bounded variance when taking expectation over the distribution of local sample $\boldsymbol{\xi}_k$, i.e.,*

$$\mathbb{E}_{\boldsymbol{\xi}_k}[\nabla Q_k(w;\boldsymbol{\xi}_k)] = \nabla J_k(w), \quad (9)$$

$$\mathbb{E}_{\boldsymbol{\xi}_k}\|\nabla Q_k(w;\boldsymbol{\xi}_k) - \nabla J_k(w)\|^2 \le \sigma^2. \quad (10)$$

The above assumption is standard in stochastic nonconvex optimization. Furthermore, we impose an assumption on the design matrices.

Assumption 4 (Combination matrices)**.** *We assume the communication matrix $W = [W_{k\ell}] \in \mathbb{R}^{K\times K}$ is a symmetric, doubly stochastic, and primitive matrix such that: 1) $\mathcal{A},\mathcal{B},\mathcal{C} \in \mathbb{R}^{Kd\times Kd}$ are polynomial functions of $W \otimes \mathbf{I}_d$, 2) $\mathcal{A},\mathcal{C}$ are symmetric and doubly stochastic, and 3) $\mathcal{B}\mathcal{W} = 0$ if $\mathcal{W} = \mathbb{1}_K \otimes w$ for a nonzero vector w.*

The above assumption is commonly used in the FO decentralized learning algorithm [16], [17], [19], [20].

B. Theoretical analysis

1) Proof Sketch: Our analysis relies on establishing a descent relation for a carefully constructed potential function. Specifically, we introduce the following potential function:

$$\mathbf{\Omega}_i = \mathbb{E}\Big[J(\boldsymbol{w}_{c,i}) + \gamma_1\| \boldsymbol{\varepsilon}_i \|^2 + \gamma_2\| \boldsymbol{s}_i \|^2 + \gamma_3\|\boldsymbol{s}_{c,i}\|^2\Big]. \quad (11)$$

where $\gamma_1,\gamma_2,\gamma_3$ are nonnegative coefficients to be determined in the analysis to ensure a sufficient descent of $\mathbf{\Omega}_i$. Here, $\boldsymbol{s}_i \in \mathbb{R}^{Kd}$ denotes the estimation error between the accelerated ZO estimator $\mathrm{col}\{\boldsymbol{g}_{k,i}\}_{k=1}^K$ and their approximation target, $\boldsymbol{s}_{c,i} \in \mathbb{R}^d$ is the centroid of $\boldsymbol{s}_i$, and $\boldsymbol{\varepsilon}_i$ is the coupled error terms:

$$\boldsymbol{\varepsilon}_i \triangleq \frac{1}{\tau}\widehat{\mathcal{Q}}^{-1}\begin{bmatrix}\widehat{\mathcal{U}}^\top \boldsymbol{\mathcal{W}}_i \\ \widehat{\Lambda}_b^{-1}\widehat{\mathcal{U}}^\top \boldsymbol{\mathcal{Z}}_i\end{bmatrix} \in \mathbb{R}^{2(K-1)d}, \quad (12)$$

where $\tau > 0$ is a design constant, and $\widehat{\mathcal{U}}^\top, \widehat{\Lambda}_b^{-1}, \widehat{\mathcal{Q}}^{-1}$ result from the eigendecomposition of $\{\mathcal{A},\mathcal{B},\mathcal{C}\}$. $\boldsymbol{\mathcal{Z}}_i$ is a tracker of the gradient estimator given by

$$\boldsymbol{\mathcal{Z}}_i \triangleq \mu\,\mathcal{A}\,\boldsymbol{\mathcal{G}}_i + \mathcal{B}\,\boldsymbol{\mathcal{D}}_i - \mathcal{B}^2\,\boldsymbol{\mathcal{W}}_i\,. \quad (13)$$

The analysis is carried out by establishing a descent relation for each term in $\mathbf{\Omega}_i$. As an illustration, by exploiting the L-smooth property of $J(\cdot)$, we obtain the following inequality:

$$J(\boldsymbol{w}_{c,i+1}) - J(\boldsymbol{w}_{c,i}) \le -\frac{\mu}{2}\|\nabla J(\boldsymbol{w}_{c,i})\|^2 - \frac{\mu}{2}(1-L\mu)\|\boldsymbol{g}_{c,i}\|^2 + 2\mu\|\boldsymbol{s}_{c,i}\|^2 + 2\mu\delta^2 dL^2, \quad (14)$$

where $\boldsymbol{g}_{c,i} = \frac{1}{K}\sum_{k=1}^K \boldsymbol{g}_{k,i}$. After deriving descent relations for the remaining stochastic terms, we combine them to bound $\mathbf{\Omega}_{i+1} - \mathbf{\Omega}_i$. The analysis is completed by bounding the expected gradient norm through selecting the appropriate coefficients $\gamma_1,\gamma_2,\gamma_3$, together with suitable hyperparameters, such that all terms associated with $\boldsymbol{g}_{c,i}$ are controlled.

2) Main result:

Theorem 1. *Under Assumptions 1—4, choosing appropriate hyperparameters, we can establish a unified bound for **ZO-ED**, **ZO-EXTRA**, and **ZO-ATC-GT** as follows*

$$\frac{1}{T}\sum_{i=0}^{T-1}\mathbb{E}\|\nabla J(\boldsymbol{w}_{c,i})\|^2 \quad (15)$$
$$\le \mathcal{O}\left(\frac{(\mathbf{\Omega}_0 - J^*)}{\mu T} + \frac{\Delta_1}{(1-\rho)(1-x)\underline{\lambda_b^2}} + \frac{\Delta b}{\mu^2 L^2} + \frac{\Delta}{K\beta'}\right).$$

where

$$x \triangleq \mathcal{O}\Big(\frac{15\mu^2L^2}{\underline{\lambda_b^2}}\Big), \quad \beta' \triangleq p + \beta - p\beta \quad (16)$$
$$\Delta \triangleq \mathcal{O}\Big(\frac{p(2dL^2\delta^2+\sigma^2)}{B}\mathbb{I}(B<N) + \frac{(2dL^2\delta^2+\sigma^2)\beta^2}{b}\Big),$$
$$\Delta_1 \triangleq \mathcal{O}\Big(\frac{p(2dL^2\delta^2+\sigma^2)}{B}\mathbb{I}(B<N) + \frac{(2dL^2\delta^2+\sigma^2)\beta^2}{b}\Big),$$

$\rho, \underline{\lambda_b^2}$ are constants depending on the design matrix choices, and $N < \infty$ in the finite-sum setting and $N = \infty$ in the stochastic setting.

Proof. The proof is omitted due to space constraint. □

The above theorem provides a unified performance bound for **ZO-ED**, **ZO-EXTRA**, and **ZO-ATC-GT**. By specifying the combination matrices $\{\mathcal{A},\mathcal{B},\mathcal{C}\}$, different values of $\rho, \underline{\lambda_b^2}$ are obtained, which in turn characterizes the performance of the corresponding algorithms over a sparse network. The bound can be further refined by appropriate choices of algorithmic parameters. We consider two representative settings. First, in the stochastic setting, we set $p = 0$ and $b = 1$, which avoids large batch sampling. Second, in the finite sum setting, we choose $p \neq 0$ and select parameters that exploit the finite-sum structure.

Corollary 1 (**Stochastic setting**)**.** *Under Theorem 1 and assuming $W \ge 0$, if we choose $\mu = \mathcal{O}\Big(\frac{K^{2/3}}{T^{1/3}}\Big), \beta = \mathcal{O}\Big(\frac{K^{1/3}}{T^{2/3}}\Big), p = 0, b_0 = \Big(\frac{T^{1/3}}{K^{2/3}}\Big), b = 1, \delta = \mathcal{O}\Big(\frac{\sigma}{\sqrt{2dL}}\Big)$, the communication (CC) and function query complexities (FQC) of **ZO-ED/EXTRA** are given by*

$$CC = \mathcal{O}\left(\frac{\varepsilon^{-3}}{K} + \frac{\varepsilon^{-2}}{(1-\lambda)^2} + \frac{\sqrt{K}\varepsilon^{-1.5}}{(1-\lambda)^{1.5}}\right), \quad (17)$$
$$FQC = \mathcal{O}\left(\frac{d\varepsilon^{-3}}{K} + \frac{d\varepsilon^{-2}}{(1-\lambda)^2} + \frac{\sqrt{K}d\varepsilon^{-1.5}}{(1-\lambda)^{1.5}} + \frac{d\varepsilon^{-1}}{K}\right).$$

The transient time in achieving linear speedup is given by

$$\min\left\{\mathcal{O}\left(\frac{K^2}{(1-\lambda)^6}\right), \mathcal{O}\left(\frac{K^2}{(1-\lambda)^3}\right)\right\}. \quad (18)$$

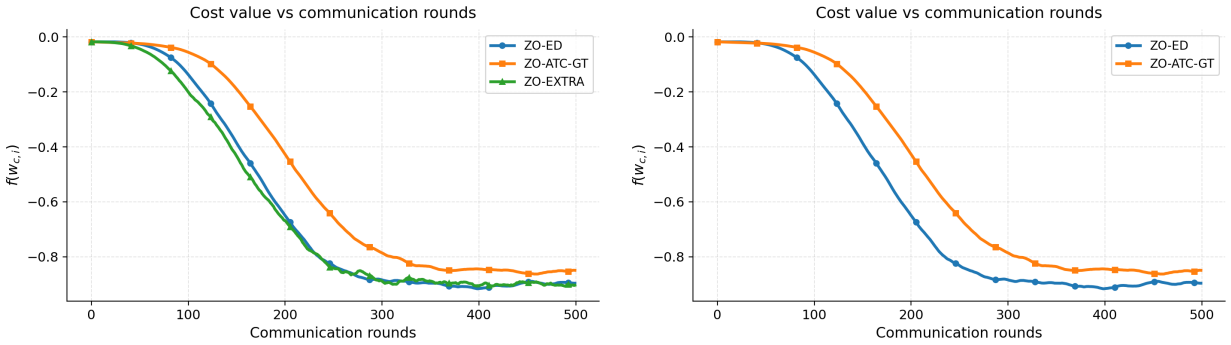
Corollary 2 (**Stochastic setting**)**.** *Under Theorem 1 and assuming $W \ge 0$, if we choose the same hyperparameters as Corollary 1, the CC and FQC of **ZO-ATC-GT** are given by*

$$CC = \mathcal{O}\left(\frac{\varepsilon^{-3}}{K} + \frac{\varepsilon^{-2}}{(1-\lambda)^3} + \frac{\sqrt{K}\varepsilon^{-1.5}}{(1-\lambda)^{9/4}}\right), \quad (19)$$
$$FQC = \mathcal{O}\left(\frac{d\varepsilon^{-3}}{K} + \frac{d\varepsilon^{-2}}{(1-\lambda)^3} + \frac{\sqrt{K}d\varepsilon^{-1.5}}{(1-\lambda)^{9/4}} + \frac{d\varepsilon^{-1}}{K}\right).$$

The transient time in achieving linear speedup is given by

$$\min\left\{\mathcal{O}\left(\frac{K^2}{(1-\lambda)^9}\right), \left(\frac{K^2}{(1-\lambda)^{4.5}}\right)\right\}. \quad (20)$$

Corollary 3 (**Finite-sum setting**)**.** *Under Theorem 1 and assuming $W \ge 0$, we choose $B = N, b = b_0 = \sqrt{N/K}, p =$*



(a) Erdős–Rényi network.

(b) Ring network.

Fig. 1: Comparison of decentralized ZO algorithm under different network topologies.

$\sqrt{1/(NK)}, \beta = 0, \mu = \mathcal{O}(1-\lambda), \delta = 1/\sqrt{dT}$ *for a sparsely connected network. Under a large* N*, the CC and FQC of* ***ZO-ED/EXTRA*** *are approximately given by*

$$CC \approx \mathcal{O}\left(\frac{\varepsilon^{-2}}{(1-\lambda)^2} + \frac{\varepsilon^{-2}}{(1-\lambda)}\right), \tag{21}$$

$$FC \approx \mathcal{O}\left(\frac{d\sqrt{N}\varepsilon^{-2}}{\sqrt{K}(1-\lambda)^2}\right). \tag{22}$$

Corollary 4 (Finite-sum setting). *Under Theorem 1 and assuming* $W \geq 0$*, we choose* $B = N, b = b_0 = \sqrt{N/K}, p = \sqrt{1/(NK)}, \beta = 0, \mu = \mathcal{O}((1-\lambda)^{1.5}), \delta = 1/\sqrt{dT}$ *for a sparsely connected network, i.e.,* $(1-\lambda) \to 0$*. Under a large* N*, the CC and FQC of* ***ZO-ATC-GT*** *are approximately given by*

$$CC \approx \mathcal{O}\left(\frac{\varepsilon^{-2}}{(1-\lambda)^3} + \frac{\varepsilon^{-2}}{(1-\lambda)^{1.5}}\right), \tag{23}$$

$$FC \approx \mathcal{O}\left(\frac{d\sqrt{N}\varepsilon^{-2}}{\sqrt{K}(1-\lambda)^3}\right). \tag{24}$$

In the stochastic setting, all algorithmic instances achieve linear speedup with respect to the K. In a finite-sum setting where full-batch evaluation is allowed, the hyperparameter settings in Corollaries 3 and 4 facilitate faster convergence speed with fewer communication rounds.

IV. Numerical Simulations

We consider a numerical example using synthesized data to evaluate the performance of the proposed ZO algorithms under different network topologies. We focus on the stochastic setting, where full-batch gradient evaluations are not available. The nonconvex risk function used in this example is given by

$$\min_{w\in\mathbb{R}^d} J(w) \triangleq \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{\xi}_k}\left[Q_k(w;\boldsymbol{\xi}_k)\right], \tag{25}$$

$$\text{where} \quad Q_k(w;\boldsymbol{\xi}_k) = \sum_{j=1}^{d}\Big((w_j - a_{kj} - \boldsymbol{\xi}_{kj})^4 - (w_j - a_{kj} - \boldsymbol{\xi}_{kj})^2\Big), \tag{26}$$

where constants $a_k = \mathrm{col}\{a_{kj}\}_{j=1}^d \in \mathbb{R}^d$ captures model heterogeneity across agents, and $\boldsymbol{\xi}_k = \mathrm{col}\{\boldsymbol{\xi}_{kj}\}_{j=1}^d \in \mathbb{R}^d$ denotes an i.i.d. stochastic perturbation. In the simulations, the parameters a_{ij} are drawn independently from $\mathrm{Unif}[-0.1, 0.1]$, and at each communication round i, each agent independently samples $\boldsymbol{\xi}_{kj} \sim \mathcal{N}(0, 0.3^2\mathbf{I}_d)$. We set the problem dimension to $d = 5$ and the number of agents to $K = 50$. We only use stochastic ZO oracles $Q_k(w;\boldsymbol{\xi}_k)$ for performing the updates.

We consider two network topologies: a ring network and an Erdős–Rényi random network using the Metropolis-Hastings rule. We set the Bernoulli parameter to $p = 0$ and the ZO smoothing parameter to $\delta = 0.001$. The remaining hyperparameters are chosen as $\beta = 0.1, \mu = 0.02$. For a fair comparison, **ZO-ED**, **ZO-EXTRA**, and **ZO-ATC-GT** use the same hyperparameter settings. All agent iterates are initialized to the same value from $N(0, 0.1^2 I_d)$.

We plot the cost value evaluated at the network centroid $\boldsymbol{w}_{c,i}$ over communication rounds. Figures (1a) and (1b) show the results over the Erdős–Rényi and ring networks, respectively. From Fig. (1a), we observe that **ZO-ED** and **ZO-EXTRA** achieve better performance than **ZO-ATC-GT** in an Erdős–Rényi network. In contrast, under a ring network, **ZO-EXTRA** diverges and is therefore omitted from Fig. (1b). These observations are consistent with findings in the FO setting regarding the impact of network topology [19], [20]. Overall, **ZO-ED** demonstrates better performance among the compared methods.

V. Conclusion

In this work, we developed a decentralized ZO framework for nonconvex gradient-free optimization. We adopt a coordinate-wise ZO strategy and incorporate the GRACE strategy to accelerate the resulting algorithm. A unified performance bound is established for a broad class of decentralized ZO methods. The theoretical results indicate that **ZO-ED** achieves better function query complexity over sparsely connected networks. Moreover, numerical experiments demonstrate that **ZO-ED** exhibits greater stability than **ZO-EXTRA**. Future work includes applying the proposed algorithm to solve practical information-theoretic applications.

REFERENCES

[1] P. Chen, H. Zhang, Y. Sharma, J. Yi, and C. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM workshop on Artificial Intelligence and Security*, pp. 15–26, 2017.

[2] S. Liu, P.-Y. Chen, B. Kailkhura, G. Zhang, A. O. Hero III, and P. K. Varshney, "A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications," *IEEE Signal Processing Magazine*, vol. 37, no. 5, pp. 43–54, 2020.

[3] C. Gratton, N. K. Venkategowda, R. Arablouei, and S. Werner, "Privacy-preserved distributed learning with zeroth-order optimization," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 265–279, 2021.

[4] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *arXiv:2001.08361*, 2020.

[5] S. Malladi, T. Gao, E. Nichani, A. Damian, J. D. Lee, D. Chen, and S. Arora, "Fine-tuning language models with just forward passes," in *NeurIPS*, vol. 36, pp. 53038–53075, 2023.

[6] S. Liu, B. Kailkhura, P.-Y. Chen, P. Ting, S. Chang, and L. Amini, "Zeroth-order stochastic variance reduction for nonconvex optimization," *NeurIPS*, vol. 31, 2018.

[7] K. Ji, Z. Wang, Y. Zhou, and Y. Liang, "Improved zeroth-order variance-reduced algorithms for nonconvex optimization," in *Proc. ICML*, pp. 3100–3109, 2019.

[8] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527–566, 2017.

[9] S. Liu, J. Chen, P.-Y. Chen, and A. Hero, "Zeroth-order online alternating direction method of multipliers: Convergence analysis and applications," in *AISTATS*, pp. 288–297, PMLR, 2018.

[10] J. Chen, S. Liu, and P.-Y. Chen, "Zeroth-order diffusion adaptation over networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4324–4328, IEEE, 2018.

[11] M. Zhang, D. Jin, J. Chen, and J. Ni, "Zeroth-order diffusion adaptive filter over networks," *IEEE Transactions on Signal Processing*, vol. 69, pp. 589–602, 2020.

[12] Y. Tang, J. Zhang, and N. Li, "Distributed zero-order algorithms for nonconvex multiagent optimization," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 1, pp. 269–281, 2020.

[13] X. Yi, S. Zhang, T. Yang, and K. H. Johansson, "Zeroth-order algorithms for stochastic distributed nonconvex optimization," *Automatica*, vol. 142, p. 110353, 2022.

[14] E. Mhanna and M. Assaad, "Single point-based distributed zeroth-order optimization with a non-convex stochastic objective function," in *International Conference on Machine Learning*, pp. 24701–24719, PMLR, 2023.

[15] H. Chen, J. Chen, and K. Wei, "A zeroth-order variance-reduced method for decentralized stochastic non-convex optimization," *Optimization*, pp. 1–43, 2025.

[16] S. A. Alghunaim and K. Yuan, "Unified and refined convergence analysis for non-convex decentralized learning," *IEEE Trans. Signal Process.*, vol. 70, pp. 3264–3279, 2022.

[17] A. H. Sayed, *Inference and Learning from Data*. Cambridge University Press, 2022.

[18] S. A. Alghunaim, E. K. Ryu, K. Yuan, and A. H. Sayed, "Decentralized proximal gradient algorithms with linear convergence rates," *IEEE Trans. Autom. Control*, vol. 66, no. 6, pp. 2787–2794, 2021.

[19] H. Cai, S. A. Alghunaim, and A. H. Sayed, "DAMA: A unified accelerated approach for decentralized nonconvex minimax optimization-part I: Algorithm development and results," *arXiv:2512.13920*, 2025.

[20] H. Cai, S. A. Alghunaim, and A. H. Sayed, "DAMA: A unified accelerated approach for decentralized nonconvex minimax optimization-part II: Convergence and performance analyses," *arXiv:2512.13923*, 2025.

[21] X. Gao, B. Jiang, and S. Zhang, "On the information-adaptive variants of admm: An iteration complexity perspective," *J. Sci. Comput.*, vol. 76, no. 1, pp. 327–363, 2018.

[22] A. Cutkosky and F. Orabona, "Momentum-based variance reduction in non-convex SGD," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019.

[23] Z. Li, H. Bao, X. Zhang, and P. Richtárik, "PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization," in *ICML*, pp. 6286–6295, PMLR, 2021.

[24] M. Liu, Z. Yuan, Y. Ying, and T. Yang, "Stochastic AUC maximization with deep neural networks," *arXiv:1908.10831*, 2019.