Variance-Reduced Gradient Estimator for Nonconvex Zeroth-Order Distributed Optimization

Huaiyi Mu[1], Yujie Tang[1], and Zhongkui Li[1]

Abstract— **This paper investigates distributed zeroth-order optimization for smooth nonconvex problems. We propose a novel variance-reduced gradient estimator, which randomly renovates one orthogonal direction of the true gradient in each iteration while leveraging historical snapshots for variance correction. By integrating this estimator with gradient tracking mechanism, we address the trade-off between convergence rate and sampling cost per zeroth-order gradient estimation that exists in current zeroth-order distributed optimization algorithms, which rely on either the 2-point or $2d$-point gradient estimators. We derive a convergence rate of $\mathcal{O}(d^{\frac{5}{2}}/m)$ for smooth nonconvex functions in terms of sampling number m and agent dimension d. Numerical simulations comparing our algorithm with existing methods confirm the effectiveness and efficiency of the proposed gradient estimator.**

I. INTRODUCTION

We consider a multi-agent system with N agents, where the agents are connected by a communication network that allows them to exchange information for decentralized decision-making. The goal of this group of agents is to collaboratively solve the following consensus optimization problem in a decentralized manner:

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

Here $x \in \mathbb{R}^d$ is the global decision variable. Each function $f_i : \mathbb{R}^d \to \mathbb{R}$ represents the local objective function for agent i, known only to the agent itself; f_i is assumed to be smooth but may be nonconvex. Each agent can only use zeroth-order information of f_i during the optimization procedure.

Decentralized optimization has gained significant attention due to its wide range of applications including multi-agent system coordination [1], power systems [2], communication networks [3], etc. For smooth and convex objective functions, the decentralized gradient descent (DGD) algorithm with a decreasing step-size achieved a convergence rate of $\mathcal{O}(\frac{\log t}{\sqrt{t}})$ [4], while [5] and subsequent works proposed gradient tracking (GT) mechanisms with a fixed step-size, achieving a sublinear convergence rate of $\mathcal{O}(\frac{1}{t})$, which is comparable to centralized gradient descent method. In many real-world applications, the objective functions can be nonconvex, and distributed nonconvex optimization has applications in areas such as machine learning [6] and robotics control [7]. For smooth nonconvex functions, DGD achieves convergence to a stationary point with a rate of $\mathcal{O}(\frac{1}{\sqrt{t}})$ [8], [9], while various GT methods can achieve convergence to a stationary point with a rate of $\mathcal{O}(\frac{1}{t})$ [10].

The aforementioned optimization algorithms rely on first-order information. However, in some scenarios, the gradient is unavailable or expensive to obtain, and agents can only access zeroth-order information of the objective functions. Such scenarios arise in optimization with black-box models [11], optimization with bandit feedback [12], fine-tuning language models [13], etc. To address this issue, various gradient-free methods, particularly algorithms based on zeroth-order gradient estimators, have attracted considerable attention [14]. The work [15] investigated the 2-point zeroth-order gradient estimator, which produces a biased stochastic gradient by using the function values of two randomly sampled points. In [16], the $2d$-point gradient estimator was proposed, where d is the dimension of the state variable for each agent. [17] combined the 2-point gradient estimator with DGD and the $2d$-point gradient estimator with GT for nonconvex multi-agent optimization, which lead to convergence rates that are comparable with their first-order counterparts. However, [17] also argued that there appears to be a trade-off between the *convergence rate* and the *sampling cost per zeroth-order gradient estimation* when combining zeroth-order gradient estimation techniques with distributed optimization frameworks. This trade-off arises from the high sampling burden of the $2d$-point estimator and the inherent variance of the 2-point estimator in distributed settings.

To address this trade-off, we aim to design a variance-reduced zeroth-order gradient estimator with a scalable sampling number of function values that is independent of the dimension d. Variance reduction is widely applied in machine learning [18] and stochastic optimization [19]. In [20], variance reduction was used for centralized stochastic gradient descent with strongly convex objectives, achieving a linear convergence rate. [21] applied a 2-point gradient estimator and employed variance reduction for zeroth-order nonconvex centralized stochastic optimization, achieving sublinear convergence. Note that these works only focused on centralized problems. Decentralized finite-sum minimization problems were considered in [22], where variance reduction was used to accelerate convergence; we note that in this work, the variance reduction technique was employed to reduce the variance caused by the finite-sum structure, rather than to address the inherent variance of the 2-point zeroth-order gradient estimator.

In this paper, we propose a new distributed zeroth-order optimization method that incorporates variance reduction

[1] The authors are with the College of Engineering, Peking University, China (e-mail: huaiyi.mu@stu.pku.edu.cn, yujietang@pku.edu.cn, zhongkli@pku.edu.cn).

techniques as well as the gradient tracking framework, to address the trade-off between *convergence rate* and *sampling cost per zeroth-order gradient estimation* in existing zeroth-order distributed optimization algorithms. Specifically, We employ the variance reduction (VR) mechanism to design a novel variance-reduced gradient estimator for distributed nonconvex zeroth-order optimization problems, as formulated in (1). We then combine this new zeroth-order gradient estimation method with the gradient tracking framework, and the resulting algorithm is able to achieve both fast convergence and low sampling cost per zeroth-order gradient estimation. To the best of the authors' knowledge, this is the first work that attempts to address the aforementioned trade-off for general zeroth-order distributed optimization problems. We also provide rigorous convergence analysis of our proposed algorithm under the smoothness assumption. Although the derived oracle complexities have a higher dependence on the dimension d compared to the algorithms in [17], numerical experiments demonstrate that our proposed algorithm enjoys superior convergence speed and accuracy, reaching lower optimization errors with the same number of samples compared to existing zeroth-order distributed optimization algorithms [17], [23].

Notations: The set of positive integers up to m is denoted as $[m] = \{1, 2, \cdots, m\}$. The i-th component of a vector x is denoted as $[x]_i$. The spectral norm and spectral radius of a matrix A are represented by $\sigma(A)$ and $\rho(A)$, respectively. For a vector $x \in \mathbb{R}^d$, $\|x\|$ refers to the Euclidean norm, and $\|x\|_\infty^\pi = \max_i |[x]_i| / [\pi]_i$ refers to the weighted infinity norm, where $\pi \in \mathbb{R}^d$ has all positive components. For a matrix A, $\|A\|_2$ represents the matrix norm induced by $\|\cdot\|$, and $|||A|||_\infty^\pi$ represents the matrix norm induced by $\|\cdot\|_\infty^\pi$. For two matrices M and N, $M \otimes N$ denotes the Kronecker product. We denote $\mathbb{B}_d$ as the closed unit ball in $\mathbb{R}^d$, and $\mathbb{S}_{d-1} = \{x \in \mathbb{R}^d : \|x\| = 1\}$ as the unit sphere. $\mathcal{U}(\cdot)$ denotes the uniform distribution.

II. Formulation And Preliminaries

A. Problem Formulation

We consider a network consisting of N agents connected via an undirected communication network. The topology of the network is represented by the graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ and $\mathcal{E}$ represent the set of agents and communication links, respectively. The distributed consensus optimization problem (1) can be equivalently reformulated as follows:

$$\begin{aligned} \min_{x_1,\ldots,x_N \in \mathbb{R}^d} \quad & \frac{1}{N}\sum_{i=1}^N f_i(x_i) \\ \text{s.t.} \quad & x_1 = x_2 = \cdots = x_N, \end{aligned} \tag{2}$$

where $x_i \in \mathbb{R}^d$ now represents the local decision variable of agent i, and the constraint $x_1 = \cdots = x_N$ requires the agents to achieve global consensus for the final decision. During the optimization procedure, each agent may obtain other agents' information only via exchanging messages with their neighbors in the communication network. We further impose the restriction that only zeroth-order information of the local objective function is available to each agent. In other words, in each iteration, agent i can query the function values of f_i at finitely many points.

The following assumption will be employed later in this paper.

Assumption 1. *Each $f_i : \mathbb{R}^d \to \mathbb{R}$ is L-smooth, i.e., we have*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \le L\|x - y\| \tag{3}$$

for all $x, y \in \mathbb{R}^d$ and $i = 1, \ldots, N$. Furthermore, $f^* = \inf_{x\in\mathbb{R}^d} f(x) > -\infty$.

B. Preliminaries on Distributed Zeroth-Order Optimization

When gradient information of the objective function is unavailable, one may construct gradient estimators by sampling the function values at a finite number of points, which has been shown to be a very effective approach by existing literature. We first briefly introduce two types of gradient estimators [17] that are commonly used in noiseless distributed optimization.

Let $h : \mathbb{R}^d \to \mathbb{R}$ be a continuously differentiable function. One version of the 2-point zeroth-order gradient estimator for $\nabla h(x)$ has the following form:

$$G_h^{(2)}(x, u, z) = d \cdot \frac{h(x + uz) - h(x - uz)}{2u} z, \tag{4}$$

where u is a positive scalar called the *smoothing radius* and z is a random vector sampled from the distribution $\mathcal{U}(\mathbb{S}_{d-1})$. One can show that the expectation of the 2-point gradient estimator is the gradient of a smoothed version of the original function [24], [25], i.e.,

$$\mathbb{E}_{z\sim\mathcal{U}(\mathbb{S}_{d-1})}[G_h^{(2)}(x, u, z)] = \nabla h^u(x),$$

where $h^u(x) = \mathbb{E}_{y\sim\mathcal{U}(\mathbb{B}_d)}[h(x + uy)]$. As the smoothing radius u tends to zero, the expectation of the 2-point gradient estimator approaches to the true gradient $\nabla h(x)$.

By combining the simple 2-point gradient estimator (4) with the decentralized gradient descent framework, one obtains the following algorithm for distributed zeroth-order consensus optimization (2):

$$x_i^{k+1} = \sum_{j=1}^N W_{ij}\left(x_j^k - \eta_t\, G_{f_j}^{(2)}(x_j^k, u^k, z_j^k)\right), \tag{5}$$

which we shall call DGD-2p in this paper. Here x_i^k denotes the local decision variable of agent i at the k-th iteration, $W \in \mathbb{R}^{N\times N}$ is a weight matrix that is taken to be doubly stochastic, and η_k is the step-size at iteration k. Since each construction of the 2-point gradient estimator (4) requires sampling only two function values, we can see that DGD-2p can achieves low sampling cost per zeroth-order gradient estimation. However, as shown by [17], DGD-2p achieves a relatively slow convergence rate $O(\sqrt{d/m}\log m)$, where m denotes the number of function value queries. [17] argued that this slow convergence rate is mainly due to the inherent variance of the 2-point gradient estimator, bounded by

$$\mathbb{E}_{z\sim\mathcal{U}(\mathbb{S}_{d-1})}\left[\|G_h^{(2)}(x, u, z)\|^2\right] \lesssim d\|\nabla h(x)\|^2 + u^2L^2d^2$$

under the assumption that function h is L-smooth. In a distributed optimization algorithm, each agent's local gradient $\nabla f_i(x_i^k)$ does not vanish to zero even if the system has reached consensus and optimality. Consequently, the inherent variance of 2-point gradient estimator is inevitable and will considerably slow down the convergence rate.

To achieve higher accuracy for zeroth-order gradient estimation, existing literature has also proposed the $2d$-point gradient estimator:

$$G_h^{(2d)}(x,u) = \sum_{l=1}^{d} \frac{h(x+ue_l) - h(x-ue_l)}{2u} e_l. \tag{6}$$

Here $e_l \in \mathbb{R}^d$ is the l-th standard basis vector such that $[e_l]_j = 1$ when $j = l$ and $[e_l]_j = 0$ otherwise. It can be shown that $\|G_h^{(2d)}(x,u) - \nabla h(x)\| \le \frac{1}{2} uL\sqrt{d}$ when h is L-smooth (see, e.g., [17]). Consequently, if we assume the function values of h can be obtained accurately and machine precision issues in numerical computations are ignored, then the $2d$-point gradient estimator can achieve arbitrarily high accuracy when approximating the true gradient. By combining (6) with the distributed gradient tracking method, one obtains the following algorithm:

$$\begin{aligned} x_i^{k+1} &= \sum_{j=1}^{N} W_{ij}\left(x_j^k - \eta\, s_j^k\right), \\ s_i^{k+1} &= \sum_{j=1}^{N} W_{ij}\left(s_j^k + G_{f_j}^{(2d)}(x_j^{k+1}, u^{k+1}) - G_{f_j}^{(2d)}(x_j^k, u^k)\right), \end{aligned} \tag{7}$$

which we shall call GT-$2d$. Here the auxiliary state variable s_j^k in (7) tracks the global gradient across iterations. Distributed zeroth-order optimization algorithms that utilize the $2d$-point gradient estimator, such as GT-$2d$, can achieve faster convergence due to precise estimation of the true gradients that allows further incorporation of gradient tracking techniques. However, GT-$2d$ has higher sampling cost per gradient estimation compared to DGD-2p: As shown in (6), $2d$ points need to be sampled for each construction of the gradient estimator. This high sampling cost may lead to poor scalability when the dimension d is large.

We remark that the $2d$-point gradient estimator (6) can also be interpreted as the expectation of the following coordinate-wise gradient estimator:

$$G_h^{(c)}(x,u,l) = d \cdot \frac{h(x+ue_l) - h(x-ue_l)}{2u} e_l, \quad l \in [d], \tag{8}$$

and we have

$$G_h^{(2d)}(x,u) = \mathbb{E}_{l\sim\mathcal{U}[d]}\left[G_f^{(c)}(x,u,l)\right], \tag{9}$$

where $\mathcal{U}[d]$ denotes the discrete uniform distribution over the set $\{1,\ldots,d\}$. The coordinate-wise gradient estimator in (8) shares the similar structure with the 2-point gradient estimator in (4). The key difference is that in (8), we restrict the perturbation direction ue_l to lie in the d orthogonal directions associated with the standard basis, instead of uniformly sampled from the unit sphere.

III. Our Algorithm

To address the trade-off between convergence rate and sampling cost per gradient estimation in zeroth-order distributed optimization, we employ a variance reduction mechanism [20] to design an improved gradient estimator. The intuition is to combine the best of both worlds, i.e., the precise approximation feature of the $2d$-point gradient estimator and the low-sampling feature of the 2-point gradient estimator.

Let k denote the iteration number. For each agent, we keep a snapshot point $\tilde{x}_i^k$ together with its associate smoothing radius $\tilde{u}_i^k$ at which we have conducted relatively accurate gradient estimation. We then use a random variable ζ_i^k generated from the Bernoulli distribution $\mathrm{Ber}(p)$ as an activation indicator for the update of the snapshot point $\tilde{x}_i^k$: When $\zeta_i^k = 1$, agent i updates the snapshot point to be the current iterate, i.e., $\tilde{x}_i^k = x_i^k$ and $\tilde{u}_i^k = u_i^k$, and takes a snapshot of the $2d$-point gradient estimator $G_{f_i}^{(2d)}(\tilde{x}_i^k, \tilde{u}_i^k)$ which provides a more accurate gradient estimation at iteration k. When $\zeta_i^k = 0$, the snapshot point $\tilde{x}_i^k$ together with the smoothing radius $\tilde{u}_k^i$ remain unchanged from the previous iteration, i.e., $\tilde{x}_i^k = \tilde{x}_i^{k-1}, \tilde{u}_i^k = \tilde{u}_i^{k-1}$.

In each iteration, agent i constructs the coordinate-wise gradient estimators $G_{f_i}^{(c)}(x_i^k, u_i^k, l_i^k)$ and $G_{f_i}^{(c)}(\tilde{x}_i^k, \tilde{u}_i^k, l_i^k)$ at the two points x_i^k and $\tilde{x}_i^k$. We then propose the following variance-reduced gradient estimator (VR-GE):

$$\begin{aligned} g_{f_i}^k = {} & G_{f_i}^{(c)}(x_i^k, u_i^k, l_i^k) - G_{f_i}^{(c)}(\tilde{x}_i^k, \tilde{u}_i^k, l_i^k) \\ & + G_{f_i}^{(2d)}(\tilde{x}_i^k, \tilde{u}_i^k), \end{aligned} \tag{10}$$

where $l_i^k \in [d]$ is randomly selected by each agent i at each iteration k. By (9), we can see that

$$\begin{aligned} & \mathbb{E}\left[g_{f_i}^k \mid x_i^k, \tilde{x}_i^k\right] \\ &= \mathbb{E}\left[G_{f_i}^{(c)}(x_i^k, u_i^k, l_i^k) \mid x_i^k, \tilde{x}_i^k\right] - \mathbb{E}\left[G_{f_i}^{(c)}(\tilde{x}_i^k, \tilde{u}_i^k, l_i^k) \mid x_i^k, \tilde{x}_i^k\right] \\ & \quad + G_{f_i}^{(2d)}(\tilde{x}_i^k, \tilde{u}_i^k) \\ &= G_{f_i}^{(2d)}(x_i^k, u_i^k), \end{aligned} \tag{11}$$

i.e., VR-GE provides a stochastic gradient of f_i with a bias $\|G_{f_i}^{(2d)}(x_i^k, u_i^k) - \nabla f_i(x_i^k)\| \le \frac{1}{2} u_i^k L\sqrt{d}$, assuming f_i is L-smooth.

The expected number of function value samples required per construction of VR-GE is $4 + 2dp$. When $p = \frac{C}{2d}$ for some absolute constant C, this becomes $4 + C$ which is independent of the dimension d. This gives VR-GE the potential to decrease the sampling cost in large-dimensional zeroth-order distributed optimization by appropriately adjusting the probability p. In the following section, specifically in Lemma 1, we will rigorously analyze the variance of VR-GE and demonstrate its variance reduction property.

In designing our distributed zeroth-order optimization algorithm, we further leverage the gradient tracking mechanisms. Existing literature (including [5], [26], [27], etc.) has demonstrated that gradient tracking mechanisms help mitigate the gap in the convergence rates between distributed optimization and centralized optimization when the objective function is smooth. Drawing inspiration from this advantage, we incorporate the variance-reduced gradient estimator with gradient tracking mechanism to design our algorithm.

The details of the proposed algorithm are outlined in Algorithm 1. Here $\alpha > 0$ is the step-size; Steps 1 and 6 implement the gradient tracking mechanism, while Steps 2–5 implement our proposed variance-reduced gradient estimator (10). The convergence guarantees of Algorithm 1 will be provided and discussed in the next section.

Algorithm 1 Distributed Zeroth-Order Optimization Algorithm with Variance Reduced Gradient Tracking Estimator

```
Initialization : $\tilde{x}_i^0 = x_i^0 \in \mathbb{R}^d, s_i^0 = g_{f_i}^0 = 0$.
for $k = 0, 1, 2, \cdots$ do
    for each $i \in [N]$ do
    1. Update $x_i^{k+1}$ by
        $$x_i^{k+1} = \sum_{j=1}^{N} W_{ij}(x_j^k - \alpha s_j^k).$$
    2. Select $l_i^{k+1}$ uniformly at random from $[d]$.
    3. Generate $\xi_i^{k+1} \sim \mathrm{Ber}(p)$.
    4. If $\zeta_i^{k+1} = 1$, update $\tilde{x}_i^{k+1} = x_i^{k+1}$, $\tilde{u}_i^{k+1} = u_i^{k+1}$;
       If $\zeta_i^{k+1} = 0$, update $\tilde{x}_i^{k+1} = \tilde{x}_i^k$, $\tilde{u}_i^{k+1} = \tilde{u}_i^k$.
    5. Construct the VR-GE $g_{f_i}^{k+1}$ by
        $$g_{f_i}^{k+1} = G_{f_i}^{(c)}(x_i^{k+1}, u_i^{k+1}, l_i^{k+1}) - G_{f_i}^{(c)}(\tilde{x}_i^{k+1}, \tilde{u}_i^{k+1}, l_i^{k+1}) + G_{f_i}^{(2d)}(\tilde{x}_i^{k+1}, \tilde{u}_i^{k+1}).$$
    6. Update $s_i^{k+1}$ by
        $$s_i^{k+1} = \sum_{j=1}^{N} W_{ij}(s_j^k + g_{f_j}^{k+1} - g_{f_j}^k).$$
    end
end
```

IV. Main Results

In this section, we present the convergence result of Algorithm 1 under Assumption 1. Due to space limit, we only provide proof sketches of Theorem 1 in Section V and refer to [28] for complete proofs.

For the subsequent analysis, we denote

$$x^k = \begin{bmatrix} x_1^k \\ \vdots \\ x_N^k \end{bmatrix}, \quad \tilde{x}^k = \begin{bmatrix} \tilde{x}_1^k \\ \vdots \\ \tilde{x}_N^k \end{bmatrix}, \quad s^k = \begin{bmatrix} s_1^k \\ \vdots \\ s_N^k \end{bmatrix},$$

and define the following quantities:

$$\begin{aligned} \delta^k &= \mathbb{E}[f(\bar{x}^k)] - f^*, & E_x^k &= \mathbb{E}[\|x^k - \mathbf{1}_N \otimes \bar{x}^k\|^2], \\ E_{\tilde{x}}^k &= \mathbb{E}[\|\tilde{x}^k - \mathbf{1}_N \otimes \bar{x}^k\|^2], & E_s^k &= \mathbb{E}[\|s^k - \mathbf{1}_N \otimes \bar{g}^k\|^2], \end{aligned}$$

where $\bar{x}^k = \frac{1}{N}\sum_{i=1}^N x_i^k$ and $\bar{g}^k = \frac{1}{N}\sum_{i=1}^N g_{f_i}^k$. Here, δ^k quantifies the optimality gap in terms of the objective value, E_x^k and $E_{\tilde{x}}^k$ characterize the consensus errors, and E_s^k characterizes the tracking error. We also denote $\tilde{u}^k = \max\{\tilde{u}_i^k\}, i \in [N]$ and $\sigma = \|W - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\|_2$.

Theorem 1. *Under Assumption 1, suppose the parameters of Algorithm 1 satisfy $p \in \left(\frac{1-\sigma^2}{d}, 1\right]$, $\sum_{\tau=0}^{\infty}(du_i^\tau)^2 < \infty$, u_i^k is non-increasing, and*

$$\alpha L \le \min\left\{\frac{1}{6\sqrt{d}}, \frac{1}{12\sqrt{d}}\left(\sqrt{\frac{2+\sigma^2}{1-p}} - \sqrt{3}\right), \frac{(1-\sigma^2)^3}{216\sqrt{29}d^{\frac{5}{2}}}\right\}.$$

Then $\lim_{k\to\infty}\mathbb{E}[f(\bar{x}^k)]$ exists,

$$\frac{1}{k}\sum_{\tau=0}^{k-1}\mathbb{E}[\|\nabla f(\bar{x}^\tau)\|^2] \le \frac{1}{k}\left(\frac{6}{\alpha}\delta^0 + \frac{6LR_0}{\sqrt{29d}N\alpha} + 36L^2R_u\right),$$

and

$$\begin{aligned} &\frac{1}{k}\sum_{\tau=0}^{k-1}\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[\|x_i^\tau - \bar{x}^\tau\|^2] \\ &\le \frac{1}{k}\left(\frac{4\sqrt{29}\delta^0}{(1-\sigma^2)L\sqrt{d}} + \frac{8R_0}{(1-\sigma^2)Nd} + \frac{7R_u}{54d^2}\right), \\ &\frac{1}{k}\sum_{\tau=0}^{k-1}\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[\|s_i^\tau - \nabla f(\bar{x}^\tau)\|^2] \\ &\le \frac{1}{k}\left(\frac{580\delta^0}{(1-\sigma^2)^2\alpha} + \frac{40\sqrt{29}LR_0}{(1-\sigma^2)N\alpha\sqrt{d}} + \frac{\sqrt{29}LR_u}{10\alpha d^{\frac{3}{2}}}\right), \end{aligned}$$

where $R_0 = \frac{d}{1-\sigma^2}E_x^0$, $R_u = \sum_{\tau=0}^{\infty}(d\tilde{u}^\tau)^2$.

Remark 1. The convergence rate of Algorithm 1 under Assumption 1 is $\mathcal{O}(\frac{1}{k})$, which aligns with the rate achieved for distributed nonconvex optimization with gradient tracking using first-order information [10]. In addition, each iteration of VR-GE requires $4+2dp$ function value queries on average. As long as $p < 1 - \frac{2}{d}$, the averaged sampling number for VR-GE is less then that for the $2d$-point gradient estimator.

Remark 2. Assuming $p \propto \frac{1}{d}$ and that all agents start from the same initial point, the convergence rate becomes $\mathcal{O}\big(d^{\frac{5}{2}}/m\big)$ with respect to the number of function value queries m, which can be justified by simple algebraic calculation. Although this rate is not as favorable as the $\mathcal{O}(d/m)$ rate of GT-$2d$ in terms of the dependence on the problem dimension d, we suspect that this discrepancy is primarily due to the complicated analysis procedure; the conditions in Theorem 1 serve as sufficient but not necessary conditions for convergence, meaning that the $\mathcal{O}\big(d^{\frac{5}{2}}/m\big)$ rate may not be a theoretically tight bound. As demonstrated in the simulation section, Algorithm 1 converges faster than GT-$2d$ and achieves higher accuracy with the same number of samples. Deriving a tighter convergence result remains a topic for future work.

Next, we present the convergence rate when the probability p is fixed as $1/d$ in the following corollary.

Corollary 1. *Under Assumption 1, suppose the parameters of Algorithm 1 satisfy $p = 1/d$, $\sum_{\tau=0}^{\infty}(du_i^\tau)^2 < \infty$, u_i^k is* *non-increasing, and*

$$\alpha L \leq \min\left\{\frac{1}{6\sqrt{d}}, \frac{\sqrt{3}}{12\sqrt{d}}\left(-1+\sqrt{1+\frac{\sigma^2}{d-1}}\right), \frac{(1-\sigma^2)^3}{264\sqrt{29}d^{\frac{3}{2}}}\right\}.$$

Then $\lim_{k\to\infty}\mathbb{E}[f(\bar{x}^k)]$ *exists, and*

$$\frac{1}{k}\sum_{\tau=0}^{k-1}\mathbb{E}[\|\nabla f(\bar{x}^\tau)\|^2] \leq \frac{1}{k}\left(\frac{6}{\alpha}\delta^0 + \frac{6\sqrt{d}L\tilde{R}_0}{\sqrt{29}N\alpha} + 36L^2R_u\right), \tag{12}$$

where $\tilde{R}_0 = \frac{1}{1-\sigma^2}E_x^0$.

Remark 3. With some standard algebraic calculation, it can be shown that the convergence rate in (12) is $\mathcal{O}\big(d^{\frac{3}{2}}/m\big)$ in terms of the number of function value queries m, which improves upon the $\mathcal{O}\big(d^{\frac{5}{2}}/m\big)$ rate in Theorem 1. Roughly speaking, this improvement is due to that, by fixing $p = 1/d$, the step-size's dependency on the dimension can be decreased from $1/d^{\frac{5}{2}}$ to $1/d^{\frac{3}{2}}$. In light of this improvement, setting $p = 1/d$ offers practical guidance for selecting the probability p.

V. Outline of Convergence Analysis

A. Bounding the Variance of VR-GE

The variance of VR-GE is essential for convergence proof of Algorithm 1 and we provide analysis details in this subsection. We first rewrite Algorithm 1 as follows:

$$x^{k+1} = (W\otimes I_d)(x^k - \alpha s^k), \tag{13a}$$

$$s^{k+1} = (W\otimes I_d)(s^k + g^{k+1} - g^k), \tag{13b}$$

where $g^k = [(g_{f_1}^k)^T, (g_{f_2}^k)^T, \cdots, (g_{f_N}^k)^T]^T$.

We now derive a bound on the expected difference between variance-reduced gradient estimator and the true gradient in the following lemma.

Lemma 1. *Let $h : \mathbb{R}^d \to \mathbb{R}$ be L-smooth. Then for any $x, \tilde{x}, y \in \mathbb{R}^d$ and $0 < u \leq \tilde{u}$, it holds that*

$$\begin{aligned}&\mathbb{E}_{l\sim\mathcal{U}[d]}\left[\|g-\nabla h(x)\|^2\right]\\ &\leq 12dL^2\|x-y\|^2 + 12dL^2\|\tilde{x}-y\|^2 + \frac{7}{2}\tilde{u}^2L^2d^2,\end{aligned} \tag{14}$$

where $g = G_h^{(c)}(x,u,l) - G_h^{(c)}(\tilde{x},\tilde{u},l) + G_h^{(2d)}(\tilde{x},\tilde{u})$.

Proof. Using $\|a-b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, we derive that

$$\begin{aligned}&\mathbb{E}_{l\sim\mathcal{U}[d]}\left[\|g-\nabla h(x)\|^2\right]\\ &= \mathbb{E}_{l\sim\mathcal{U}[d]}\Big[\Big\|\left(G_h^{(c)}(x,u,l) - G_h^{(c)}(\tilde{x},\tilde{u},l)\right)\\ &\quad -\left(G_h^{(2d)}(x,u)-G_h^{(2d)}(\tilde{x},\tilde{u})\right) - \left(\nabla h(x) - G_h^{(2d)}(x,u)\right)\Big\|^2\Big]\\ &\leq 2\mathbb{E}_{l\sim\mathcal{U}[d]}\Big[\Big\|\left(G_h^{(c)}(x,u,l) - G_h^{(c)}(\tilde{x},\tilde{u},l)\right)\\ &\quad - \left(G_h^{(2d)}(x,u) - G_h^{(2d)}(\tilde{x},\tilde{u})\right)\Big\|^2\Big]\\ &\quad + 2\mathbb{E}[\|\nabla h(x) - G_h^{(2d)}(x,u)\|^2].\end{aligned}$$

Then, by $\mathbb{E}[\|\zeta - \mathbb{E}[\zeta]\|^2] \leq \mathbb{E}[\|\zeta\|^2]$ for an arbitrary random vector ζ, we can derive from the above inequality that

$$\begin{aligned}&\mathbb{E}_{l\sim\mathcal{U}[d]}\left[\|g-\nabla h(x)\|^2\right]\\ &\leq 2\mathbb{E}_{l\sim\mathcal{U}[d]}\left[\|G_h^{(c)}(x,u,l) - G_h^{(c)}(\tilde{x},\tilde{u},l)\|^2\right] + \frac{1}{2}u^2L^2d\\ &= 2d\|G_h^{(2d)}(x,u) - G_h^{(2d)}(\tilde{x},\tilde{u})\|^2 + \frac{1}{2}u^2L^2d\\ &= 2d\|(G_h^{(2d)}(x,u) - \nabla h(x)) - (G_h^{(2d)}(\tilde{x},\tilde{u}) - \nabla h(\tilde{x}))\\ &\quad + (\nabla h(x) - \nabla h(\tilde{x}))\|^2 + \frac{1}{2}u^2L^2d\\ &\leq 6dL^2\|x-\tilde{x}\|^2 + \frac{3}{2}u^2L^2d^2 + \frac{3}{2}\tilde{u}^2L^2d^2 + \frac{1}{2}u^2L^2d\\ &\leq 12dL^2\|x-y\|^2 + 12dL^2\|\tilde{x}-y\|^2 + \frac{7}{2}\tilde{u}^2L^2d^2,\end{aligned}$$

where we have used (3) and inequality $\|G_h^{(2d)}(x,u) - \nabla h(x)\| \leq \frac{1}{2}uL\sqrt{d}$ in the second inequality, and the fact that $u \leq \tilde{u}$ in the last inequality. □

Remark 4. The estimation error of VR-GE (i.e., the left-hand side of (14)) comprises two parts: consensus error and the smoothing radius effect. Consensus error is inherent due to the distributed nature of the algorithm, while the smoothing radius can be tuned properly. As Algorithm 1 approaches consensus and the smoothing radius approaches zero, the estimation error between the VR-GE and the true gradient diminishes. Thus, VR-GE offers reduced variance compared to the 2-point gradient estimator while requiring fewer samples than the $2d$-point gradient estimator.

B. Proof Sketch of Theorem 1

The proof relies on four lemmas. The first lemma analyzes the evolution of function value $f(\bar{x}^k)$ using L-smooth property. Without loss of generality, we assume $d \geq 3$ in the following analysis.

Lemma 2. *Suppose $\alpha L \leq \frac{1}{6\sqrt{d}}$, we have*

$$\begin{aligned}\delta^{k+1} \leq \delta^k - \frac{\alpha}{3}\mathbb{E}[\|\nabla f(\bar{x}^k)\|^2] + \frac{18\alpha dL^2}{N}E_x^k\\ + \frac{16\alpha dL^2}{N}E_{\tilde{x}}^k + 5\alpha L^2d^2(\tilde{u}^k)^2.\end{aligned} \tag{15}$$

Lemma 2 derives a bound for optimization error δ^k. We need to further bound consensus errors: E_x^k and $E_{\tilde{x}}^k$. This is tackled by the following lemma.

Lemma 3. *Suppose we choose $p \in \left(\frac{1-\sigma^2}{d}, 1\right]$ and $\alpha L \leq \min\left\{\frac{(1-\sigma^2)^2}{12\sigma^2(1+2\sigma^2)\sqrt{d(1+p)}}, \frac{1}{6\sqrt{d}}, \frac{\sqrt{3}}{12d}\left(-\sqrt{d}+\sqrt{\frac{d-1+\sigma^2}{1-p}}\right)\right\}$. Then we have the following inequality:*

$$v^{k+1} \leq Av^k + B_k, \tag{16}$$

where

$$v^k = \left[E_x^k, E_{\tilde{x}}^k, \frac{\alpha}{3\sqrt{29}L\sqrt{d}}E_s^k\right]^T,$$

$$A = \begin{bmatrix}\frac{1+2\sigma^2}{3} & 0 & \frac{9\sqrt{29}\sqrt{d}}{1-\sigma^2}\alpha L\\ 17\sqrt{d}\alpha L + \frac{1+2\sigma^2}{3} & 1-\frac{1-\sigma^2}{d} & \frac{9\sqrt{29}\sqrt{d}}{1-\sigma^2}\alpha L\\ \frac{9\sqrt{29}\sqrt{d}}{1-\sigma^2}\alpha L & \frac{3\sqrt{29}\sqrt{d}}{1-\sigma^2}\alpha L & \frac{2+\sigma^2}{3}\end{bmatrix},$$

$$\textit{and } B_k = \begin{bmatrix}0\\ \frac{N\alpha}{L\sqrt{d}}\mathbb{E}[\|\nabla f(\bar{x}^k)\|^2] + 5Nd^{\frac{3}{2}}\alpha L(\tilde{u}^k)^2\\ \frac{\sqrt{29}N\alpha}{3(1-\sigma^2)L\sqrt{d}}\mathbb{E}[\|\nabla f(\bar{x}^k)\|^2] + \frac{2\sqrt{29}Nd^{\frac{3}{2}}\alpha L}{1-\sigma^2}(\tilde{u}^k)^2\end{bmatrix}.$$

For subsequent analysis, we need to bound the induced weighted matrix norm of the matrix A in the inequality (16).

Lemma 4. *Consider the matrix A defined in Lemma 3, and suppose that $\alpha L \leq \frac{(1-\sigma^2)^3}{12\sqrt{29}d^{\frac{5}{2}}}$. Then*

$$\rho(A) \leq |||A|||_{\infty}^{\pi} \leq 1 - \frac{1-\sigma^2}{2d}, \tag{17}$$

where $\pi = [\frac{1-\sigma^2}{d}, 3, 1]^T$.

Next, we derive a bound on the accumulated consensus errors over iterations using Lemma 3.

Lemma 5. *Suppose we choose $p \in (\frac{1-\sigma^2}{d}, 1]$ and $\alpha L \leq \min\left\{\frac{1}{6\sqrt{d}}, \frac{(1-\sigma^2)^3}{12\sqrt{29}d^{\frac{5}{2}}}, \frac{\sqrt{3}}{12d}\left(-\sqrt{d} + \sqrt{\frac{d-1+\sigma^2}{1-p}}\right)\right\}$. We have*

$$\begin{aligned}
&\max\left\{\frac{d}{1-\sigma^2}\sum_{\tau=0}^{k} E_x^{\tau}, \frac{1}{3}\sum_{\tau=0}^{k} E_{\tilde{x}}^{\tau}, \frac{\alpha}{3\sqrt{29}L\sqrt{d}}\sum_{\tau=0}^{k} E_s^{\tau}\right\} \\
&\leq \frac{4d}{1-\sigma^2}R_0 + \frac{2\sqrt{29}\sqrt{d}N\alpha}{3(1-\sigma^2)^2 L}\sum_{m=0}^{k-1}\mathbb{E}[\|\nabla f(\bar{x}^m)\|^2] \\
&\quad + \frac{4\sqrt{29}Nd^{\frac{5}{2}}\alpha L}{(1-\sigma^2)^2}\sum_{m=0}^{k-1}(\tilde{u}^m)^2,
\end{aligned} \tag{18}$$

where $R_0 = \frac{d}{1-\sigma^2}E_x^0$.

Based on Lemmas 2 and 5, we are now ready to prove Theorem 1. Plugging the inequality (18) into (15) leads to

$$\begin{aligned}
0 &\leq \delta^0 - \frac{\alpha}{3}\sum_{\tau=0}^{k}\mathbb{E}[\|\nabla f(\bar{x}^{\tau})\|^2] + 5\alpha L^2 d^2 \sum_{\tau=0}^{k}(\tilde{u}^{\tau})^2 \\
&\quad + \frac{54\alpha d L^2}{N}(\frac{4d}{1-\sigma^2}R_0 + \frac{4\sqrt{29}Nd^{\frac{5}{2}}\alpha L}{(1-\sigma^2)^2}\sum_{m=0}^{k-1}(\tilde{u}^m)^2 \\
&\quad + \frac{2\sqrt{29}\sqrt{d}N\alpha}{3(1-\sigma^2)^2 L}\sum_{m=0}^{k-1}\mathbb{E}[\|\nabla f(\bar{x}^m)\|^2]) \\
&\leq \delta^0 + \frac{L}{\sqrt{29}N\sqrt{d}}R_0 - \frac{\alpha}{6}\sum_{\tau=0}^{k}\mathbb{E}[\|\nabla f(\bar{x}^{\tau})\|^2] \\
&\quad + 6\alpha L^2 d^2 \sum_{\tau=0}^{k}(\tilde{u}^{\tau})^2,
\end{aligned} \tag{19}$$

where we have used $\frac{18\alpha d L^2}{N}\cdot\frac{1-\sigma^2}{d} + \frac{16\alpha d L^2}{N}\cdot 3 < \frac{54\alpha d L^2}{N}$ in the first inequality and $\alpha L \leq \frac{(1-\sigma^2)^3}{216\sqrt{29}d^{\frac{5}{2}}} \leq \frac{1-\sigma^2}{216\sqrt{29}d^{\frac{3}{2}}}$ in the last inequality.

Since $(u_i^k)^2$ is summable, we see that $(\tilde{u}^k)^2$ is also summable. Consequently, we have $\mathbb{E}[\|\nabla f(\bar{x}^k)\|^2] \to 0$ as $k \to \infty$ by (19). We can further derive from the inequality (19) that

$$\begin{aligned}
&\frac{1}{k}\sum_{\tau=0}^{k-1}\mathbb{E}[\|\nabla f(\bar{x}^{\tau})\|^2] \\
&\leq \frac{1}{k}\left[\frac{6}{\alpha}\delta^0 + \frac{6L}{\sqrt{29}\sqrt{d}N\alpha}R_0 + 36L^2 d^2\sum_{\tau=0}^{\infty}(\tilde{u}^{\tau})^2\right].
\end{aligned} \tag{20}$$

Combining (20) with (18), we have

$$\frac{1}{N}\sum_{\tau=0}^{\infty}E_x^{\tau} \leq \frac{4\sqrt{29}\delta^0}{(1-\sigma^2)L\sqrt{d}} + \frac{8R_0}{(1-\sigma^2)Nd} + \frac{7}{54}\sum_{\tau=0}^{\infty}(\tilde{u}^{\tau})^2, \tag{21}$$

where we have used $\alpha L \leq \frac{(1-\sigma^2)^3}{216\sqrt{29}d^{\frac{5}{2}}}$. Similarly,

$$\begin{aligned}
&\frac{1}{N}\sum_{\tau=0}^{\infty}\mathbb{E}[\|s^{\tau} - \mathbf{1}_N \otimes \nabla f(\bar{x}^{\tau})\|^2] \\
&\leq \frac{3}{2N}\sum_{\tau=0}^{\infty}\mathbb{E}[\|s^{\tau} - \mathbf{1}_N \otimes \bar{g}^{\tau}\|^2] + 3\sum_{\tau=0}^{\infty}\mathbb{E}[\|\bar{g}^{\tau} - \nabla f(\bar{x}^{\tau})\|^2] \\
&\leq \frac{3}{2N}\sum_{\tau=0}^{\infty}E_s^{\tau} + 3\sum_{\tau=0}^{\infty}\left[\frac{26dL^2}{N}E_x^{\tau} + \frac{24dL^2}{N}E_{\tilde{x}}^{\tau} + 7(\tilde{u}^{\tau})^2 L^2 d^2\right] \\
&\leq \frac{5\sqrt{29}\sqrt{d}L}{N\alpha}\left(\frac{4d}{1-\sigma^2}R_0 + \frac{4\sqrt{29}Nd^{\frac{5}{2}}\alpha L}{(1-\sigma^2)^2}\sum_{m=0}^{k-1}(\tilde{u}^m)^2\right. \\
&\quad \left. + \frac{2\sqrt{29}dN\alpha}{3(1-\sigma^2)^2 L\sqrt{d}}\sum_{m=0}^{k-1}\mathbb{E}[\|\nabla f(\bar{x}^m)\|^2]\right) + 21d^2L^2\sum_{\tau=0}^{\infty}(\tilde{u}^{\tau})^2 \\
&\leq \frac{580\delta^0}{(1-\sigma^2)^2\alpha} + \frac{40\sqrt{29}LR_0}{(1-\sigma^2)N\alpha\sqrt{d}} + \frac{\sqrt{29}\sqrt{d}L}{10\alpha}\sum_{\tau=0}^{\infty}(\tilde{u}^{\tau})^2,
\end{aligned}$$

where we have used inequality (18) and $\frac{3}{2N}\cdot\frac{3\sqrt{29}L\sqrt{d}}{\alpha} + \frac{78dL^2}{N}\cdot\frac{1-\sigma^2}{d} + \frac{72dL^2}{N}\cdot 3 \leq \frac{5\sqrt{29}\sqrt{d}L}{N\alpha}$ in the third inequality. The proof of Theorem 1 can now be concluded.

VI. Simulation

We consider a multi-agent nonconvex optimization problem adapted from [17] with $N = 50$ agents in the network, and the objective function of each agent is given as follows:

$$f_i(x) = \frac{\alpha_i}{1 + e^{-\zeta_i^T x - v_i}} + \beta_i \ln(1 + \|x\|^2), \tag{22}$$

where $\alpha_i, \beta_i, v_i \in \mathbb{R}$ are randomly generated parameters satisfying $\frac{1}{N}\sum_i \beta_i = 1$, each $\zeta_i \in \mathbb{R}^d$ is also randomly generated, and the dimension d is set to 64.

For the following numerical simulation of Algorithm 1, we set the step-size $\eta = 0.02$ and the smoothing radius $u_i^k = 3/k^{\frac{3}{4}}$. All agents start from the same initial points to ensure consistency in the initial conditions across the network.

A. Comparison with Other Algorithms

Fig. 1 compares Algorithm 1 with DGD-2p, GT-2d, and ZONE-M (with J=100). In the figure, the probability used for Algorithm 1 is $p = 0.1$. The horizontal axis is normalized and represents the sampling number m (i.e., the number of zeroth-order queries). The two sub-figures illustrate the stationarity gap $\|\nabla f(\bar{x}^k)\|^2$ and the consensus error $\frac{1}{N}\sum_i \|x_i^k - \bar{x}^k\|^2$, respectively.

By inspecting Fig. 1, we first see that DGD-2p converges faster than ZONE-M with $J = 100$. When comparing DGD-2p and GT-2d, we can see a clear difference between their convergence behavior: DGD-2p achieves fast convergence initially but slows down afterwards due to the inherent variance of the 2-point gradient estimator, whereas GT-2d

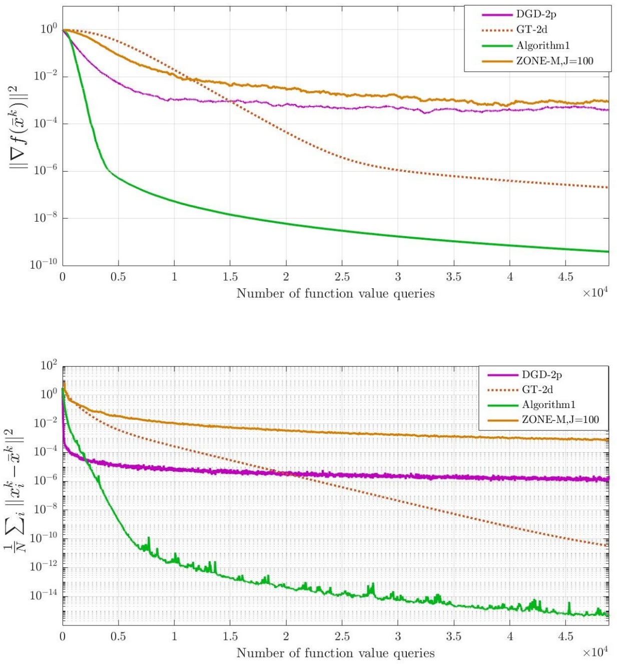

Fig. 1. Convergence of Algorithm 1, ZONE-M with J =100, DGD-2p, GT-2d.

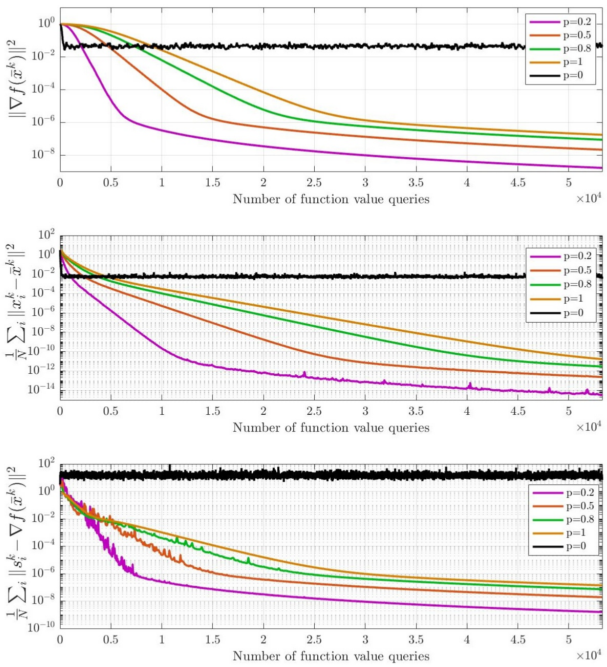

Fig. 2. Convergence of Algorithm 1 under probability p = 0.2, 0.5, 0.8, and 1.

achieves higher eventual accuracy but slower initial convergence before approximately 1.5×10^4 zeroth-order queries due to the higher sampling burden of the $2d$-point gradient estimator.

As demonstrated in Fig. 1, Algorithm 1 offers both high eventual accuracy and a fast convergence rate in terms of stationarity gap and consensus error. This improvement is attributed to the variance reduction mechanism employed in designing VR-GE, which effectively balances the sampling number and expected variance, thereby addressing the trade-off between convergence rate and sampling cost per zeroth-order gradient estimation that exists in current zeroth-order distributed optimization algorithms.

B. *Comparison of Algorithm 1 under Different Probabilities*

Fig. 2 compares the convergence of Algorithm 1 under different choices of the probability p, which reflects the frequency with which each agent takes snapshots. The three sub-figures illustrate the stationarity gap $\|\nabla f(\bar{x}^k)\|^2$, the consensus error $\frac{1}{N}\sum_i \|x_i^k - \bar{x}^k\|^2$, and the tracking error $\frac{1}{N}\sum_i \|s_i^k - \nabla f(\bar{x}^k)\|^2$, respectively.

The results demonstrate that Algorithm 1 with a lower probability achieves better accuracy with fewer sampling numbers. However, a lower probability also results in more fluctuation during convergence. This is expected because, with a lower probability, the snapshot variables are updated less frequently, leading to a greater deviation from the true gradient as iterations progress.

Two notable cases are $p = 0$ and $p = 1$. When $p = 1$, VR-GE behaves similarly to GT-2d, taking snapshots at every iteration, which results in poorer convergence performance compared to cases where $p \in (0, 1)$. Conversely, when $p = 0$, agents do not take any snapshots, and $\tilde{x}_i^k$ remains its initial value x_i^0. In this scenario, VR-GE uses solely the initial iterate for gradient correction, which introduces variance and reduces convergence accuracy, as illustrated in Fig. 2.

C. *Comparison of Algorithm 1 under Different Dimensions*

Fig. 3 compares the convergence of Algorithm 1 across different agent dimensions, alongside varying probabilities for taking snapshots within the algorithm. The results show that Algorithm 1 can effectively handle different scenarios, such as when $d = 300$, achieving stationarity gaps that are below 10^{-6}.

As the dimension increases, VR-GE requires more samples to accurately estimate the gradient. To maintain similar convergence performance across higher dimensions, the probability p for taking snapshots can be adjusted to lower values. As shown in Fig. 3, decreasing the probability as the dimension grows allows Algorithm 1 to achieve a convergence rate and optimization accuracy that are comparable to cases with lower dimensions. However, this adjustment also leads to increased fluctuation during the convergence process. This fluctuation is a result of the randomness introduced by the snapshot mechanism.

VII. Conclusion

In this paper, we proposed an improved variance-reduced gradient estimator and integrated it with gradient tracking mechanism for nonconvex distributed zeroth-order optimization problems. Through rigorous analysis, we demonstrated that our algorithm achieves sublinear convergence for smooth nonconvex functions that is comparable with first-order gradient tracking algorithms, while maintaining relatively low

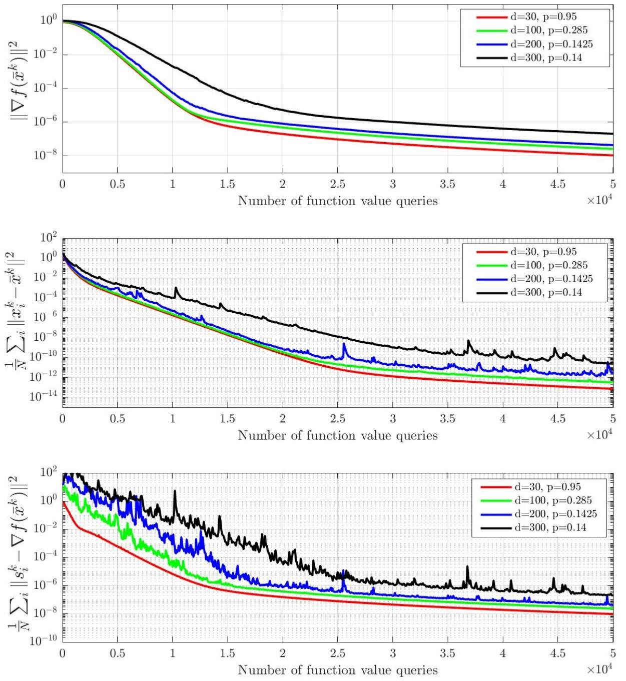

Fig. 3. Convergence of Algorithm 1 with different dimension d = 30, 100, 200, and 300.

sampling cost per gradient estimation. Comparative evaluations with existing distributed zeroth-order optimization algorithms verified the effectiveness of the proposed gradient estimator. Future work will focus on refining the step-size bounds of the algorithm and reducing the dependence of the convergence rate on the dimension d.

References

[1] Shu Liang, Le Yi Wang, and George Yin. Distributed smooth convex optimization with coupled constraints. *IEEE Transactions on Automatic Control*, 65(1):347–353, 2020.

[2] Xuan Zhang, Antonis Papachristodoulou, and Na Li. Distributed control for reaching optimal steady state in network systems: An optimization approach. *IEEE Transactions on Automatic Control*, 63(3):864–871, 2018.

[3] Jie Liu, Daniel W. C. Ho, and Lulu Li. A generic algorithm framework for distributed optimization over the time-varying network with communication delays. *IEEE Transactions on Automatic Control*, 69(1):371–378, 2024.

[4] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[5] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

[6] Angelia Nedic. Distributed gradient methods for convex machine learning problems in networks: Distributed optimization. *IEEE Signal Processing Magazine*, 37(3):92–101, 2020.

[7] Guido Carnevale, Nicola Mimmo, and Giuseppe Notarstefano. Nonconvex distributed feedback optimization for aggregative cooperative robotics. *Automatica*, 167:111767, 2024.

[8] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *Advances in neural information processing systems*, 30, 2017.

[9] Jinshan Zeng and Wotao Yin. On nonconvex decentralized gradient descent. *IEEE Transactions on signal processing*, 66(11):2834–2848, 2018.

[10] Gesualdo Scutari and Ying Sun. Distributed nonconvex constrained optimization over time-varying digraphs. *Mathematical Programming*, 176:497–544, 2019.

[11] Zhongguo Li, Zhen Dong, Zhongchao Liang, and Zhengtao Ding. Surrogate-based distributed optimisation for expensive black-box functions. *Automatica*, 125:109407, 2021.

[12] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[13] Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075, 2023.

[14] Anit Kumar Sahu, Dusan Jakovetic, Dragana Bajovic, and Soummya Kar. Communication-efficient distributed strongly convex stochastic optimization: Non-asymptotic rates. *arXiv preprint arXiv:1809.02920*, 2018.

[15] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

[16] Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, pages 462–466, 1952.

[17] Yujie Tang, Junshan Zhang, and Na Li. Distributed zero-order algorithms for nonconvex multiagent optimization. *IEEE Transactions on Control of Network Systems*, 8(1):269–281, 2021.

[18] Yongyi Guo, Dominic Coey, Mikael Konutgan, Wenting Li, Chris Schoener, and Matt Goldman. Machine learning for variance reduction in online experiments. *Advances in Neural Information Processing Systems*, 34:8637–8648, 2021.

[19] Chong Wang, Xi Chen, Alexander J Smola, and Eric P Xing. Variance reduction for stochastic gradient optimization. *Advances in neural information processing systems*, 26, 2013.

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

[21] Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[22] Ran Xin, Usman A Khan, and Soummya Kar. Variance-reduced decentralized stochastic optimization with accelerated convergence. *IEEE Transactions on Signal Processing*, 68:6255–6271, 2020.

[23] Davood Hajinezhad, Mingyi Hong, and Alfredo Garcia. Zone: Zeroth-order nonconvex multiagent optimization over networks. *IEEE transactions on automatic control*, 64(10):3995–4010, 2019.

[24] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience, 1983.

[25] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394, 2005.

[26] Guannan Qu and Na Li. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3):1245–1260, 2017.

[27] Angelia Nedic, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017.

[28] Huaiyi Mu, Yujie Tang, and Zhongkui Li. Variance-reduced gradient estimator for nonconvex zeroth-order distributed optimization, 2024. Technical report. Available at `https://tyj518.github.io/files/Zeroth-Order-VR-GE.pdf`.