Stochastic Zeroth-order Optimization via Variance Reduction method

Liu Liu*[1], Minhao Cheng†[2], Cho-Jui Hsieh‡[2], and Dacheng Tao§[1]

[1]UBTECH Sydney AI Centre and SIT, FEIT, The University of Sydney
[2]University of California, Davis

Abstract

Derivative-free optimization has become an important technique used in machine learning for optimizing black-box models. To conduct updates without explicitly computing gradient, most current approaches iteratively sample a random search direction from Gaussian distribution and compute the estimated gradient along that direction. However, due to the variance in the search direction, the convergence rates and query complexities of existing methods suffer from a factor of d*, where* d *is the problem dimension. In this paper, we introduce a novel Stochastic Zeroth-order method with Variance Reduction under Gaussian smoothing (SZVR-G) and establish the complexity for optimizing non-convex problems. With variance reduction on both sample space and search space, the complexity of our algorithm is sublinear to* d *and is strictly better than current approaches, in both smooth and non-smooth cases. Moreover, we extend the proposed method to the mini-batch version. Our experimental results demonstrate the superior performance of the proposed method over existing derivative-free optimization techniques. Furthermore, we successfully apply our method to conduct a universal black-box attack to deep neural networks and present some interesting results.*

1. Introduction

Derivative-free optimization methods have a long history in optimization [1]. They use only function value information rather than explicit gradient calculation to optimize a function, as in the case of black-box setting or when computing the partial derivative is too expensive. Recently, derivative-free methods received substantial attention in machine learning and deep learning [2], such as online problem in bandit setting [3, 4, 5], certain graphical model and structure-prediction problems [6], and black-box attack to deep neural networks (DNNs) [7, 8, 9]. However, the convergence rate of current approaches encounters a factor of d, where d is problem dimension. This prevents the application of derivative-free optimization in high-dimensional problems.

This paper focuses on the theoretical development of derivative-free (zeroth-order) method for non-convex optimization. More specifically, we consider the following optimization problem:

$$\min_{x\in\mathbb{R}^d} f(x) \overset{\text{def}}{=} \frac{1}{n}\sum\nolimits_{i=1}^{n} F(x;\xi_i), \tag{1.1}$$

where $f(x)$ and $F(x,\xi_i) : \mathbb{R}^d \to \mathbb{R}$ are differentiable, non-convex functions, and ξ_i, $i \in [n]$ is a random variable. In particular, when n=1, the objective function is $f(x)$=$F(x,\xi)$ with a fixed ξ, which becomes the problem solved in [10]. To solve (1.1), most approaches [10] consider the use of stochastic zeroth-order oracle ($\mathcal{SZO}$). At each iteration, for a given x, u and ξ, $\mathcal{SZO}$ outputs a stochastic gradient $G_\mu(x,u,\xi)$ defined by

$$G_\mu\left(x,\xi,u\right) = \frac{F\left(x+\mu u,\xi\right) - F\left(x,\xi\right)}{\mu}u, \tag{1.2}$$

*lliu8101@uni.sydney.edu.au
†mhcheng@ucdavis.edu
‡chohsieh@ucdavis.edu
§dacheng.tao@sydney.edu.au

Table 1. Comparison of $\mathcal{SZO}$ complexity for the non-convex problem

Method	$\mathcal{SZO}$ complexity	Mini-Batch	Non-smooth
RGF [10]	$O\left(\frac{dn}{\varepsilon^2}\right)$	$O\left(\frac{dn}{\varepsilon^2}b_0\right)$	$O\left(\frac{d^5 n}{\varepsilon^5}\right)$
RSG [11]	$O\left(\frac{d}{\varepsilon^2}+\frac{\sqrt{d}}{\varepsilon^4}\right)$	$O\left(\left(\frac{d}{\varepsilon^2}+\frac{\sqrt{d}}{\varepsilon^4}\right)b_0\right)$	-
SZVR-G	$O\left(\max\left\{\frac{d^{\frac{2}{3}}B^{\frac{1}{3}}}{\varepsilon^2},\frac{d^{\frac{1}{3}}B^{\frac{2}{3}}}{\varepsilon^2}\right\}\right)$	$O\left(\max\left\{\frac{(db_0)^{\frac{2}{3}}B^{\frac{1}{3}}}{\varepsilon^2},\frac{(db_0)^{\frac{1}{3}}B^{\frac{2}{3}}}{\varepsilon^2}\right\}\right)$	$O\left(\frac{d^{\frac{5}{3}}B^{\frac{1}{3}}}{\varepsilon^{\frac{11}{3}}}\right)$

which approximates the derivative along the direction of u. Each $\mathcal{SZO}$ only requires 2 function value evaluations (or 1 if $F(x,\xi)$ has already being queried). It is thus natural to analyze the convergence rate of an algorithm in terms of number of $\mathcal{SZO}$ required to achieve $\|\nabla f(x)\|^2 \leq \epsilon^2$ with a small ϵ.

A recent important work by Nesterov and Spokoiny [10] proposed the random gradient-free method (RGF) and proved some tight bounds for approximation the gradient through function value information with Gaussian smoothing techniques. He established an $O(d/\varepsilon^2)$ complexity for non-convex smooth function in the case of n=1 in problem (1.1). Subsequently, Ghadimi and Lan [11] introduced a randomized stochastic gradient (RSG) method for solving the stochastic programming problem (1.2) and proved the complexity of $O(d/\varepsilon^2+\sqrt{d}/\varepsilon^4)$. However, when the dimension d is large, especially in deep learning, these derivative-free methods will suffer slow convergence.

The dependency in d is mainly due to the variance in sampling query direction u. Recently, a family of variance reduction methods have been proposed for first-order optimization, including SVRG [12], SCSG [13] and Natasha [14]. They developed ways to reduce variance of stochastic samples (ξ). It is thus natural to ask the following question: can the variance reduction technique also be used in derivative-free optimization to reduce the $\mathcal{SZO}$ complexity caused by problem dimension? And how to choose the best size of Gaussian random vector set for each epoch to estimate the gradient in zeroth-order optimization?

In this paper, we develop a novel stochastic zeroth-order method with variance reduction under Gaussian smoothing (SZVR-G). The main contributions are summarized below.

- We proposed a novel algorithm based on variance reduction. Different from RSG and RGF that generate a Gaussian random vector for each iteration, we independently generate Gaussian vector set (in practice, we preserve the corresponding seeds) to compute the average of direction derivatives at the beginning of each epoch as defined in (3.1). In the inner iteration of epoch, we randomly select one or block of seeds that preserved in the outer epoch to compute the corresponding gradient as defined in (3.2).

- We give the theoretical proof for the proposed algorithm and show that our results are better than that of RGF and RSG in both smooth and non-smooth functions, and in the case of both $n=1$ and $n\neq 1$ of problem (1.1). Furthermore, we also explicitly present parameter settings and the corresponding derivation process, which is better for understanding the convergence analysis.

- We extend the stochastic zeroth-order optimization to the mini-batch setting. Although the $\mathcal{SZO}$ complexity will increase, we show that the increasing rate is sublinear to batch size. In comparison, previous algorithms including RGF and RSG have complexity growing linearly with batch size. Furthermore, the total number of iterations in our algorithm will decrease when using larger mini-batch, which implicitly implies better parallelizability.

- We show that our algorithm is more efficient than both RGF and RSG in canonical logistic regression problem. Furthermore, we successfully apply our algorithm to a real black-box adversarial attack problem that involves high-dimensional zeroth order optimization.

1.1. Our results

Our proposed algorithm can achieve the following $T_{\mathcal{SZO}}$ complexity:

$$O\left(\frac{1}{\varepsilon^2}\max\{d^{2/3}B^{1/3},d^{1/3}B^{2/3}\}\right),$$

where $B=\min\{n,1/\varepsilon^2\}$. We identify an interesting dichotomy with respect to d. In particular, if $d\geq B$, $T_{\mathcal{SZO}}$ becomes $O(d^{2/3}B^{1/3}/\varepsilon^2)$, otherwise $T_{\mathcal{SZO}}$ becomes $O(d^{1/3}B^{2/3}/\varepsilon^2)$. Different complexities of methods are presented in Table 1.

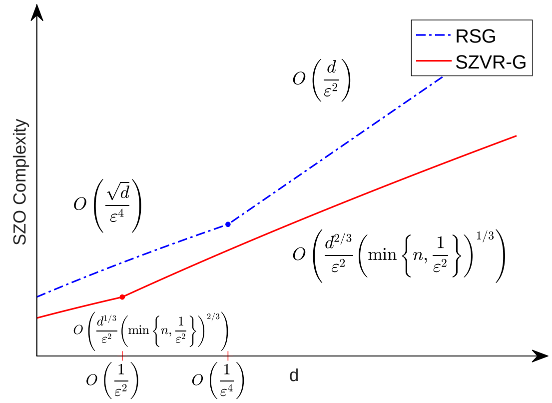

Figure 1. The SZO complexities of RSG and SZVR-G with different dimension d. Note that in the plot we assume n is infinity (we set $B = 1/\varepsilon^2$).

Comparing our method with RGF [10] in the case of n=1 (that is $B = 1$), we can see that our result is better than that of RGF with a factor of $d^{1/3}$ improvement. For $n > 1$, the complexity of our method is also better than that of RSG [11] as clearly shown in Figure 1.

Mini-Batch Our result generalizes to the mini-batch stochastic setting, where in the inner iteration of each epoch, the estimated gradient $\tilde{\nabla}_k$ defined in (3.2) is computed with mini-batch of b_0 times. The $\mathcal{SZO}$ complexity will become $O(\frac{1}{\varepsilon^2}\max\{d^{2/3}B^{1/3}b_0^{2/3}, d^{1/3}B^{2/3}b_0^{1/3}\})$. The comparison of mini-batch $\mathcal{SZO}$ complexity is also shown in Table 1.

Non-smooth We also give the convergence analysis for non-smooth case and present the $\mathcal{SZO}$ complexity, which is better than that of RGF [10].

1.2. Other Related work

Derivative-free optimization can be dated back to the early days of the development of the optimization theory [1]. The advantage of using derivative-free method is manifested in the case when computation of function value is much simpler than gradient, or in the black-box setting when optimizer does not have full information about the function.

The most common method for derivative-free optimization is the random optimization approach [1], which samples a random vector uniformly distributed over the unit sphere, computes the directional derivative of the function, and then moves the next point if the update leads to the decrease of function value. However, no particular convergence rate was established. Nesterov and Spokoiny [9] presented several random derivative-free methods, and provide the corresponding complexity bound for both convex and non-convex problems. What's more, an important kind of smoothness, Gaussian smoothing and its properties were established. Ghadimi and Lan [11] incorporated the Gaussian smoothing technique to randomized stochastic gradient (RSG). John *et al.* [5, 15] analyzed the finite-sample convergence rate of zeroth-order optimization for convex problem. Wang *et al.* [16] considered the zeroth-order optimization in high-dimension, but also in convex function. For the coordinate smoothness (the sampled direction is along natural basis), Lian *et al.* [17] presented zeroth-order under asynchronous stochastic parallel optimization for non-convex problem. Subsequently, Gu *et al.* [18] apply variance reduction of zeroth-order to asynchronous doubly stochastic algorithm, however, without the specific analysis of the complexity related to dimension d. Furthermore, it is not practical to perform full gradient computation in the parallel setting for large-scale data.

Stochastic first-order methods including SGD [19] and SVRG [20] have been studied extensively. However, these two algorithms suffer from either hight iteration complexity or the complexity that depend on the number of samples. Lei *et al.* [13] recently proposed the stochastically controlled stochastic gradient (SCSG) method to obtain the complexity that is based on $\min\{n, 1/\varepsilon^2\}$, which is derived from [21] and [22] for the convex case.

The rest of the paper is organized as following. We first introduce some notations, definitions and assumptions in Section 2. In Section 3, we provide our algorithm via variance reduction technology, and analyze the complexity for both smooth and non-smooth function, and their corresponding mini-batch version. Experiment results are shown in 4. Section 5 concludes our paper.

2. Preliminary

Throughout this paper, we use Euclidean norm denoted by $\|\cdot\|$. We use $i \in [n]$ to denote that i is generated from $[n] = \{1, 2, ..., n\}$. We denote by $\mathcal{B}$ and $\mathcal{D}$ the set, and $B = |\mathcal{B}|$ and $D = |\mathcal{D}|$ the cardinality of the sets. We use $\xi_{\mathcal{B}}$ and $u_{\mathcal{D}}$ to denote the variable set, where $\xi_{\mathcal{B}[b]}$ belong to $\xi_{\mathcal{B}}$, $b \in \mathcal{B}$, and $u_{\mathcal{D}[j]}$ belong to $u_{\mathcal{D}}$, $j \in \mathcal{D}$. We use $\mathbb{I}[event]$ to denote the indicator function of a probabilistic event. Here are some definitions on the smoothness of a function, direction derivative and smooth approximation function and its property.

Definition 2.1. *For a function $f(x) : \mathbb{R}^d \to \mathbb{R}$, $\forall x, y \in \mathbb{R}^d$,*

- *$f(x) \in C^{0,0}$, then $|f(x) - f(y)| \leq L_0 \|x - y\|$.*
- *$f(x) \in C^{1,1}$, then $\|\nabla f(x) - \nabla f(y)\| \leq L_1 \|x - y\|$ and $f(y) \leq f(x) + \langle \nabla f(x, \xi), y - x\rangle + \frac{L_1}{2}\|y - x\|^2$.*

Note that if $F(x, \xi) \in C^{1,1}$, then $f(x) \in C^{1,1}$ due to the fact that

$$\|\nabla f(x) - \nabla f(y)\| = \|\mathbb{E}_{\xi}[\nabla F(x, \xi)] - \mathbb{E}_{\xi}[\nabla F(y, \xi)]\| \leq \mathbb{E}_{\xi}[\|\nabla F(x, \xi) - \nabla F(y, \xi)\|] \leq L_1 \|x - y\|.$$

Definition 2.2. *The smooth approximation of $f(x)$ is defined as*

$$f_\mu(x) = \frac{1}{(2\pi)^{n/2}} \int f(x + \mu u) e^{\frac{1}{2}\|u\|^2} du, \mu > 0. \tag{2.1}$$

Its corresponding gradient is $\nabla f_\mu(x) = \mathbb{E}_{\xi,u}[G_\mu(x, u, \xi)]$ and $\nabla F_\mu(x, \xi) = E_u[G_\mu(x, u, \xi)]$, where $G_\mu(x, u, \xi)$ defined in (1.2). The details of gradient derivation process can be referred to [10].

Lemma 2.1. *[10] For $f_\mu(x)$ defined in (2.1),*

- *If $f \in C^{0,0}$, then $f_\mu \in C^{1,1}$ with $L_1(f_\mu) = \frac{1}{\mu} d^{1/2} L_0$, and $|f_\mu(x) - f(x)| \leq \mu L_0 d^{1/2}$.*
- *If $f \in C^{1,1}$, then $f_\mu \in C^{1,1}, L_1(f_\mu) \leq L_1(f)$, and*

$$\begin{aligned}
\|\nabla f(x)\|^2 \leq& 2\|\nabla f_\mu(x)\|^2 + \frac{1}{2}\mu^2 L_1^2 (d + 6)^3, \\
\mathbb{E}_u \|G_\mu(x, u, \xi) - \nabla f_\mu(x)\|^2 \leq& \frac{\mu^2}{2} L_1^2 (d + 6)^3 + 2(d + 4)\|\nabla f(x)\|^2.
\end{aligned}$$

Assumption 2.1. *We assume that H is the upper bound on the variance of function $\nabla F_\mu(x, \xi_i)$, that is*

$$\frac{1}{n} \sum\nolimits_{i=1}^{n} \|\nabla f_\mu(x) - \nabla F_\mu(x, \xi_i)\|^2 \leq H.$$

3. Stochastic Zeroth-order via Variance reduction with Gaussian smooth

We introduce our SZVR-G method in Algorithm 1. At each outer iteration, we have two kinds of sampling: the first one is to form $\xi_{\mathcal{B}}$ with the size of B, which are randomly selected from $[n]$; the second one is to independently generate a Gaussian vector set $u_{\mathcal{D}}$ with D times. Furthermore, we store the corresponding seeds of Gaussian vectors, which will be used for the inner iterations. The main difference between set $\mathcal{D}$ and $\mathcal{B}$ is the property of independence, which will be the key element in analyzing the size of their sets. Based on these two sets, we compute the random gradient at a snapshot point $\tilde{x}_s$, which is maintained for each epoch,

$$G_\mu(\tilde{x}_s, u_{\mathcal{D}}, \xi_{\mathcal{B}}) = \frac{1}{D} \sum\nolimits_{j=1}^{D} \frac{1}{B} \sum\nolimits_{i=1}^{B} G_\mu(\tilde{x}_s, u_{\mathcal{D}[j]}, \xi_{\mathcal{B}[i]}), \tag{3.1}$$

where the definition of $G_\mu(x, u, \xi)$ is in (1.2).

At each inner iteration, we select i and j from $[n]$ and $\mathcal{D}$ randomly, and compute the estimated random gradient,

$$\tilde{\nabla}_k = G_\mu(x_k, u_{\mathcal{D}[j]}, \xi_i) - G_\mu(\tilde{x}_s, u_{\mathcal{D}[j]}, \xi_i) + G_\mu(\tilde{x}_s, u_{\mathcal{D}}, \xi_{\mathcal{B}}), \tag{3.2}$$

where $u_{\mathcal{D}}$ and $\xi_{\mathcal{B}}$ are the Gaussian vector set and sample set. Taking expectation of $\tilde{\nabla}_k$ with respect to i, j and u, we have

$$\mathbb{E}_{i,j,u}[\tilde{\nabla}_k] = \nabla f_\mu(x_k) - \nabla f_\mu(\tilde{x}_s) + \nabla F_\mu(\tilde{x}_s, \xi_{\mathcal{B}}), \tag{3.3}$$

where $\nabla f_\mu(x)$ and $\nabla F_\mu(x, \xi)$ are defined in Definition 2.2.

Algorithm 1 Zeroth-order via variance reduction with Gaussian smooth

Require: K, S, η (learning rate), and $\tilde{x}_0$

for $s = 0, 1, 2, \cdots, S-1$ **do**

Independently Generate Gaussian vector set $u_{\mathcal{D}}$ through Gaussian random vector generator with D times, where $\mathcal{D}$ is the index set. ▷ In practice, store Gaussian random vector seeds for each sth iteration.

Sample from $[n]$ to form mini-batch $\mathcal{B}$ with $|\mathcal{B}| = B$.

$x_0 = \tilde{x}_s$

$G = G_\mu(x, u_{\mathcal{D}}, \xi_{\mathcal{B}})$ ▷ (3.1)

for $k = 0, 1, 2, \cdots, K-1$ **do**

Sample i from $[n]$ and j from $\mathcal{D}$

$\tilde{\nabla}_k = G_\mu\left(x_k, u_{\mathcal{D}[j]}, \xi_i\right) - G_\mu\left(\tilde{x}_s, u_{\mathcal{D}[j]}, \xi_i\right) + G$

$x_{k+1} = x_k - \eta\tilde{\nabla}_k$

end for

Update $\tilde{x}_{s+1} = x_K$

end for

Output: $\tilde{x}_k^s$, $s \in \{1, ..., S\}$, $k \in \{1, ..., K\}$

3.1. Convergence analysis

We present the convergence and complexity results for our algorithm. Theorem 3.1 is based on the variance reduction technique for the non-convex problem. The detailed proof can be found in Appendix B. In order to ensure the convergence, we present the parameter settings, such as c_0, q, K, w_0 and D in Remark B.2 and B.1.

Theorem 3.1. *In Algorithm 1, under Assumption 2.1, for $F(x,\xi) \in C^{1,1}$, let parameters $\mu, \eta, q, K > 0$, $c_0 \leq L_1$ and the cardinality of Gaussian vector set and sample set $D \geq O(\eta d)$, we have*

$$\frac{1}{S}\sum\nolimits_{s=1}^{S}\frac{1}{K}\sum\nolimits_{k=0}^{K-1}\left\|\nabla f\left(x_k^s\right)\right\|^2 \leq \frac{32R}{SK\eta} + \frac{32}{\eta}J_0 + \frac{1}{2}\mu^2 L_1^2 (d+6)^3,$$

where x^* is the optimal value of function $f_\mu(x)$, $R = \max_x\{f_\mu(x) - f_\mu(x_*) : f_\mu(x) \leq f_\mu(x_0)\}$, and

$$J_0 = \frac{3}{4}\left(\frac{1}{D} + \frac{\mathbb{I}(B<n)}{B} + 3\right)(L_1 + 2c_0)\,\mu L_1^2 (d+6)^3 \eta^2 + \left(1 + \frac{1}{q}c_0\right)\frac{1}{2}\eta\frac{\mathbb{I}(B<n)}{B}H. \tag{3.4}$$

The $\mathcal{SZO}$ complexity is presented in Theorem 3.2, which is based on the best choice of step size η. For the different sizes of B and d, we give different results, which is an interesting phenomenon caused by two types of samples.

Theorem 3.2. *In Algorithm 1, for $F(x,\xi) \in C^{1,1}$, let the size of sample set $\mathcal{B}$, B $=O(\min\{n, 1/\varepsilon^2\})$, the step $\eta = O(\min\{1/(d^{2/3}B^{1/3}), 1/(d^{1/3}B^{2/3})\})$, $\mu \leq O(\varepsilon/(L_1 d^{1.5}))$, and the number of inner iteration $K \leq O(1/\max\{d\eta^2, d^{0.5}\eta^{1.5}\})$, Gaussian vectors set $D \geq O(\eta d)$. In order to obtain*

$$\frac{1}{S}\sum\nolimits_{s=0}^{S-1}\frac{1}{K}\sum\nolimits_{k=0}^{K-1}\left\|\nabla f\left(x_k^s\right)\right\|^2 \leq \varepsilon^2,$$

the total number of $T_{\mathcal{SZO}}$ is at most $O(\frac{1}{\varepsilon^2}\max\left\{d^{2/3}B^{1/3}, d^{1/3}B^{2/3}\right\})$, with the number of total iterations $T = SK > O(1/(\varepsilon^2\eta))$.

3.1.1 Variance reduction for Gaussian random direction

If we only consider the directions of Gaussian random vector, that is $n = 1$, Algorithm 1 is similar to SVRG but the variance reduction will be on random directions instead of random samples. In outer iteration, we independently produce Gaussian random vectors and compute the smoothed gradient estimator $G_\mu(x, u_{\mathcal{D}}, \xi_{\mathcal{B}})$ in (3.1) (Here, we use ξ to indicate the only sample). Then in inner iteration, we randomly select a Gaussian vector, and compute the estimated gradient as $\tilde{\nabla}_k$ in (3.2). Since this is the same problem solved in Nesterov and Spokoiny [10], we compare the $\mathcal{SZO}$ complexity between our method and theirs based on different step-size choices:

- For $\eta > 1/d$, we set $\eta = 1/d^{2/3}$, the $\mathcal{SZO}$ complexity of our proposed method is $T_{\mathcal{SZO}}$ = $O(d^{2/3}/\varepsilon^2)$ which is better than that of RGF [10], $O(d/\varepsilon^2)$. The corresponding Gaussian random vector set $\mathcal{D}$, $D > O(\eta d) > 1$. This is due to the fact that $\nabla f_\mu(\tilde{x}_s)$ is not finite-sum structure and the term $\mathbb{E}_u\|G_\mu(\tilde{x}_s, u_D, \xi_\mathcal{B}) - \nabla f_\mu(\tilde{x}_s)\|^2 \neq 0$, which is bounded by $O(\mu^2 d^3 + d\|\nabla f(x)\|^2)/D$. More details can be referred to Lemma B.1. This is the key difference with SVRG method [20]. Based on the lower bound, we can derive the corresponding best complexity and best step as shown in Theorem 3.2.
- For $\eta \leq 1/d$, the $\mathcal{SZO}$ complexity will be larger than $O(d/\varepsilon^2)$. This can be directly seen from the total number of T. In this case D becomes 1, and the proposed algorithm will become the original RGF [10] method, where the step is $O(1/d)$. This can explain that why the variance reduction method is better than that of RGF, that is our proposed method can apply the large step to obtain the better complexity.

3.1.2 Variance reduction for finite-sum function

For the finite-sum function as in (1.1), In Algorithm 1, we also provide the variance-reduction technique at the same time for both Gaussian vector and random variable ξ. Our algorithm has two kinds of random procedure. That is, in outer iteration, we compute the gradient include both B samples and D Gaussian random vectors. In inner iteration, we randomly select a sample and a Gaussian random vector to estimate the gradient. Here, we compare our result with RSG [11], which also use both random sample and Gaussian random vector. Based on the result in Theorem 3.2, we discuss the $\mathcal{SZO}$ complexity under different d,

- For $d < B$, the $\mathcal{SZO}$ complexity of our proposed method is T_{SZO} =$O(d^{1/3}B^{2/3}/\varepsilon^2)$. This result is similar to SCSG [13] if the dimension d is not large enough. Furthermore, in our algorithm, we set B as the fix value rather than a value that is produced by the probability. If B=d, the complexity result looks the same as RSG [11]. But the difference lies on that the B is no more than $1/\varepsilon^2$ such that our result is better than RSG [11]. Figure 1 clearly shows the difference.
- For $d > B$, the $\mathcal{SZO}$ complexity becomes $O(d^{2/3}B^{1/3}/\varepsilon^2)$. The complexity is also better than that of RSG.

Based on above discussions, we conclude that the $\mathcal{SZO}$ complexity of our proposed method is better than that of RSG [11] and RGF [10].

3.2. Mini-batch SZVR-G

We extend the SZVR-G to the mini-batch version in Algorithm 2, which is similar to Algorithm 1. The difference is that we estimate the gradient in inner epoch with b_0 times computation, then average them. Theorem 3.3 gives the corresponding complexity and the corresponding step size.

Theorem 3.3. *In Algorithm 2, under Assumption 2.1, for $F(x,\xi) \in C^{1,1}$, let the size of the sample set $\mathcal{B}$, B =$O(\min\{n, 1/\varepsilon^2\})$, the step η =$O(\min\{b_0^{1/3}/(d^{2/3}B^{1/3}), b_0^{2/3}/(d^{1/3}B^{2/3})\})$, $\mu \leq O(\varepsilon/(L_1 d^{1.5}))$, and the number of inner iteration $K \leq O(1/\max\{d\eta^2, d^{0.5}\eta^{1.5}\})$, Gaussian vectors set $D \geq O(\eta d)$. In order to obtain*

$$\frac{1}{S}\sum\nolimits_{s=0}^{S-1}\frac{1}{K}\sum\nolimits_{k=0}^{K-1}\|\nabla f\left(x_k^s\right)\|^2 \leq \varepsilon^2,$$

the total number of $T_{\mathcal{SZO}}$ is at most $O(\max\{d^{2/3}B^{1/3}b_0^{2/3}, d^{1/3}B^{2/3}b_0^{1/3}\})$, with number of total iterations $T > O(1/(\varepsilon^2\eta))$.

From the above Theorem, we can see that the $\mathcal{SZO}$ complexity is increased by a factor $b_0^{2/3}$ or $b_0^{1/3}$, which is smaller than the size of the mini-batch. However, the corresponding complexity of RGF and RSG will be increased by multiplying a factor of b_0 (see Table 1), so our algorithm has a better dependency to the batch size. Furthermore, our total number of iterations will decrease by a factor $b_0^{2/3}$ or $b_0^{1/3}$.

3.3. SZVR-G for non-smooth function

For non-smooth function, we also provide the theory analysis and give the corresponding $\mathcal{SZO}$ complexity. Similar to Theorem 3.2, we analyze the convergence based on the norm of the gradient. But the difference lies in that the convergence of gradient norm is $\|\nabla f_\mu(x)\|^2$ rather than $\|\nabla f(x)\|^2$. As stated in [10], allowing $\eta \to 0$ and $\mu \to 0$, the convergence of $\|\nabla f_\mu(x)\|^2$ ensures the convergence to a stationary point of the initial function.

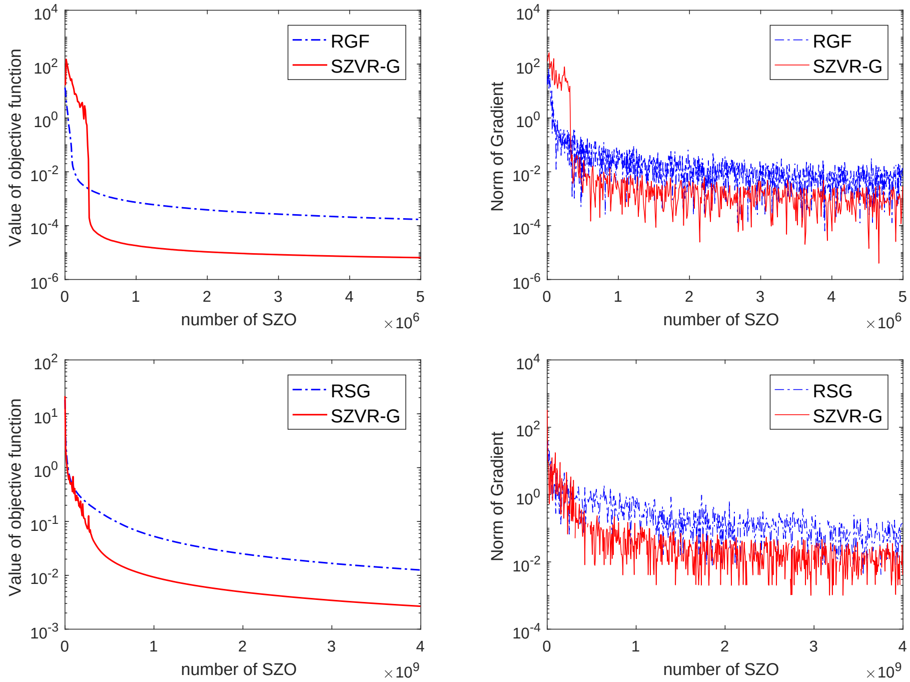

Figure 2. Comparison of different methods for the objective function value and the norm of gradient. x-axis is the number of $\mathcal{SZO}$. The first row shows the difference between SZVR-G and RGF for the case of n=100. Note that we only use $f(x)$, rather than $F(x, \xi)$ in both algorithms in order to verify the variance reduction technology in random direction vector. The second row shows the difference between SZVR-G and RSG under the condition of n=2000.

Theorem 3.4. *In Algorithm 1, for $F(x, \xi) \in C^{0,0}$, the step $\eta = O(\varepsilon^{5/3}/(d^{5/3}B^{1/3}))$, $\mu \leq O(\varepsilon/(L_0 d^{1/2}))$, and the number of inner iteration $K \leq O(\varepsilon^2/(d^2\eta^2))$, Gaussian vectors set $D \geq O(\eta d^3/\varepsilon^3)$. In order to obtain*

$$\frac{1}{S}\sum\nolimits_{s=0}^{S-1}\frac{1}{K}\sum\nolimits_{k=0}^{K-1}\left\|\nabla f_\mu\left(x_k^s\right)\right\|^2 \leq \varepsilon^2,$$

the total number of $T_{\mathcal{SZO}}$ is $O(d^{5/3}B^{1/3}/\varepsilon^{11/3})$, number of inner iterations $T > O(1/(\varepsilon^2\eta))$.

For the case of n=1, we can see that our $\mathcal{SZO}$ complexity is better than the $O(d^3/\varepsilon^5)$ complexity of RGF [10]. RSG [11] do not provide the complexity of non-smooth function. Additionally, the mini-batch version for the non-smooth function is similar to Theorem 3.3. We present the results in Theorem C.1.

4. Experimental results

4.1. Logistic regression with stochastic zeroth-order method

In order to verify our theory, we apply our algorithm to logistic regression. Given n training examples $\{(\xi_1, y_1), (\xi_2, y_2), ..., (\xi_n, y_n)\}$, where $\xi_i \in \mathbb{R}^d$ and y_i, $i \in [n]$ are the feature vector and the label of ith example. The objective function is

$$J(\theta) = -\frac{1}{n}\left[\sum\nolimits_{i=1}^{n} y^{(i)}\log h_\theta(\xi^{(i)}) + (1 - y^{(i)})\log(1 - h_\theta(\xi^{(i)}))\right],$$

where $h_\theta = 1/(1 + e^{-\theta^T\xi})$. We use MNIST [23] dataset to make two kinds of experiments in order to verify that our variance reduction technology is better than current approach. The dimension of θ is $d = 14 \times 14 \times 10$, where the size of the image is 14×14, and the number of the class is 10. We choose the parameters according to setting in Theorem 3.2 to give the best performance. First, to verify that our variance reduction technique for Gaussian random directions are useful, we compare our algorithm with RGF [10] for solving a deterministic function $f(x)$, which is the logistic regression with $n = 100$ MNIST samples. Row 1 in Figure 2 shows the results that our method SZVR-G is better than RGF [10] both on

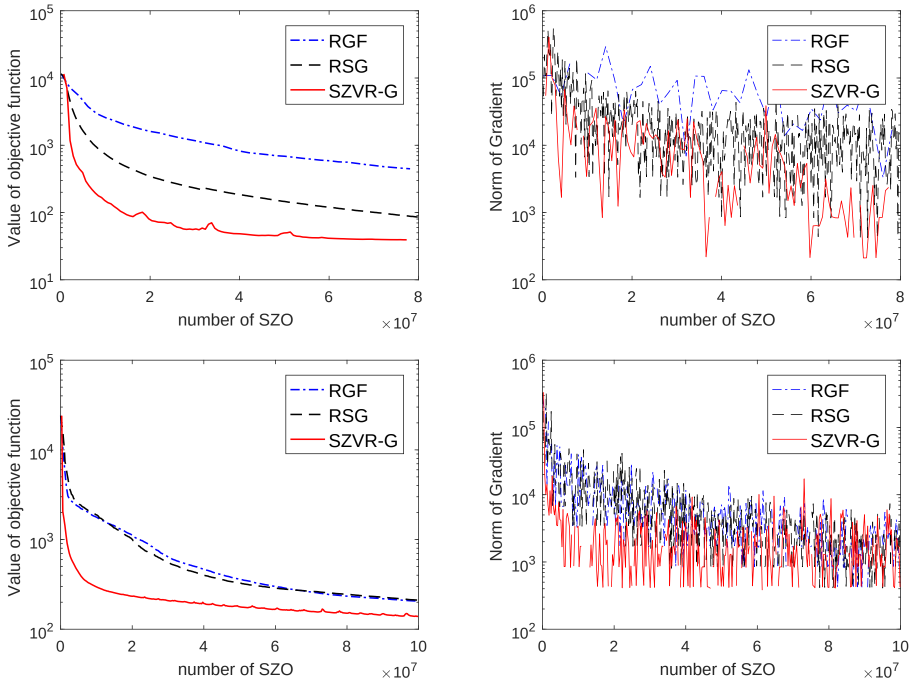

Figure 3. Comparison of RGF, RSG and our SZVR-G for the objective function value and the norm of gradient, x-axis is the number of the $\mathcal{SZO}$. Dateset: CIFAR-10 (Line 1) and MNIST (Line 2)

the objective function value and the norm of the gradient. This verified that even for solving a deterministic function, our algorithm outperforms RGF in both theory and practice, due to the variance reduction for Gaussian search directions.

In the second experiment we compare with RSG [11] on stochastic optimization, that consider two kinds of stochastic process: randomly select one or block example and Gaussian vector to estimate the gradient. We use the fix dataset with randomly selected $n = 2000$ examples. Figure 2. row 2 shows that our method is better than RSG since we conduct variance reduction on both examples and Gaussian vectors.

4.2. Universal adversarial examples with black-box setting

In the second set of experiments, we apply zeroth order optimization methods to solve a real problem in adversarial black-box attack to machine learning models. It has been observed recently that convolutional neural networks are vulnerable to adversarial example [24, 25]. [8] apply zeroth order optimization techniques in the black-box setting, where one can only acquire input-output correspondences of targeted model. Also, [26] finds there exists universal perturbations that could fool the classifier on almost all datapoints sampled. Therefore, we decide to apply our SZVR-G algorithm to non-smooth function that find universal adversarial perturbations in the black-box setting to show our efficiency in an interesting application. For classification models in neural networks, given the classification model $f : \mathbb{R}^d \to \{1, \dots, K\}$, it is usually assumed that $f(x) = \text{argmax}_i(Z(x)_i)$, where $Z(x) \in \mathbb{R}^K$ is the final layer output, and $Z(x)_i$ is the prediction score for the i-th class. Formally, we want to find a universal perturbation θ that could fool all N images in samples set $\Omega = \{(x_1, l_1), (x_2, l_2), \dots, (x_N, l_N)\}$, that is,

$$\text{argmin}_\theta\, L(\theta) = C \sum_{i=1}^{N} \max\{[Z(x_i + \theta)]_{l_i} - \max_{j \neq l_i}[Z(x_i + \theta)]_j, -\kappa\} + ||\theta||_2^2,$$

where C is a constant to balance the distortion and attack success rate and $\kappa \geq 0$ is a confidence parameter that guarantees a constant gap between $\max_{j \neq l_i}[Z(x_i+\theta)]_j$ and $[Z(x_i+\theta)]_{l_i}$. In this experiments, we use two standard datasets: MNIST [23], CIFAR-10 [27]. We construct two convolution neural networks following [28]. In detail, both MNIST and CIFAR use the same network structure with four convolution layers, two max-pooling layers and two fully-connected layers. Using the parameters provided by [28], we could achieve 99.5% accuracy on MNIST and 82.5% accuracy on CIFAR-10. All models are trained using Pytorch[1]. The dimension of θ is $d = 28 \times 28$ for MNIST and $d = 3 \times 32 \times 32$ for CIFAR-10. We tune the

[1]https://github.com/pytorch/pytorch

best parameters to give the best performance. Figure 3 show the performance with difference methods. We can see that our algorithm SZVR-G is better than RGF and RSG both on objective value and the norm of the gradient.

5. Conclusion

In this paper, we present stochastic zeroth-order optimization via variance reduction for both smooth and non-smooth nonconvex problem. The stochastic process include two kinds of aspects: randomly select the sample and derivative of direction, respectively. We give the theoretical analysis of $\mathcal{SZO}$ complexity, which is better than that of RGF and RSG. Furthermore, we also extend our algorithm to mini-batch, in which the $\mathcal{SZO}$ complexity is multiplying a smaller size of the mini-batch. Our experimental result also confirm our theory.

References

[1] J Matyas. Random optimization. *Automation and Remote control*, 26(2):246–253, 1965.

[2] Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization*, volume 8. Siam, 2009.

[3] 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. Society for Industrial and Applied Mathematics, 2005.

[4] Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, pages 28–40. Citeseer, 2010.

[5] Andre Wibisono, Martin J Wainwright, Michael I Jordan, and John C Duchi. Finite sample convergence rates of zero-order stochastic optimization methods. In *Advances in Neural Information Processing Systems*, pages 1439–1447, 2012.

[6] James C Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. John Wiley & Sons, 2005.

[7] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Exploring the space of black-box attacks on deep neural networks. *arXiv preprint arXiv:1712.09491*, 2017.

[8] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui 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*, pages 15–26. ACM, 2017.

[9] Nina Narodytska and Shiva Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1310–1318. IEEE, 2017.

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

[11] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[12] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

[13] Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via scsg methods. In *Advances in Neural Information Processing Systems*, pages 2345–2355, 2017.

[14] Zeyuan Allen-Zhu. Natasha 2: Faster non-convex optimization than sgd. *arXiv preprint arXiv:1708.08694*, 2017.

[15] John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

[16] Yining Wang, Simon Du, Sivaraman Balakrishnan, and Aarti Singh. Stochastic zeroth-order optimization in high dimensions. *arXiv preprint arXiv:1710.10551*, 2017.

[17] Xiangru Lian, Huan Zhang, Cho-Jui Hsieh, Yijun Huang, and Ji Liu. A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order. In *Advances in Neural Information Processing Systems*, pages 3054–3062, 2016.

[18] Bin Gu, Zhouyuan Huo, and Heng Huang. Zeroth-order asynchronous doubly stochastic algorithm with variance reduction. *arXiv preprint arXiv:1612.01425*, 2016.

[19] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[20] Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pages 314–323, 2016.

[21] Reza Harikandeh, Mohamed Osama Ahmed, Alim Virani, Mark Schmidt, Jakub Konečný, and Scott Sallinen. Stopwasting my gradients: Practical svrg. In *Advances in Neural Information Processing Systems*, pages 2251–2259, 2015.

[22] Lihua Lei and Michael Jordan. Less than a single pass: Stochastically controlled stochastic gradient. In *Artificial Intelligence and Statistics*, pages 148–156, 2017.

[23] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[24] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[25] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations.

[27] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[28] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 39–57. IEEE, 2017.

A. Technical Lemma

Lemma A.1. *For the sequences that satisfy $c_{k-1} = c_k Y + U$, where $Y > 1$, $U > 0$, $k \geq 1$ and $c_0 > 0$, we can get the geometric progression*

$$c_k + \tfrac{U}{Y-1} = \tfrac{1}{Y}\left(c_{k-1} + \tfrac{U}{Y-1}\right),$$

then c_k can be represented as decrease sequences,

$$c_k = \left(\tfrac{1}{Y}\right)^k \left(c_0 + \tfrac{U}{Y-1}\right) - \tfrac{U}{Y-1}.$$

Furthermore, if $c_K \geq 0$, we have

$$c_0 = \tfrac{U\left(Y^K - 1\right)}{Y-1}.$$

Lemma A.2. *[10] For $u \in \mathbb{R}^d$ and $p \geq 0$, $\frac{1}{\kappa}\int\limits_E \|u\|^p e^{-\frac{1}{2}\|u\|^2} du \leq (p+d)^{p/2}$, where $\kappa = \int\limits_E e^{-\frac{1}{2}\|u\|^2} du$.*

Lemma A.3. *[10] If $f(x)$ is differentiable at x then,*

$$E_u[\|f'(x) \cdot u\|^2] = E_u[\langle \nabla f(x), u\rangle \|u\|^2] = (d+4)\|\nabla f(x)\|^2, \tag{A.1}$$

where $f'(x) = \lim\limits_{\alpha \downarrow 0} \frac{1}{\alpha}[f(x+\alpha u) - f(x)]$.

Lemma A.4. *If $v_1, ..., v_n \in \mathbb{R}^d$ satisfy $\sum_{i=1}^n v_i = \vec{0}$, and $\mathcal{B}$ is a non-empty, uniform random subset of $[n]$, then*

$$\mathbb{E}_{\mathcal{B}}\left\|\frac{1}{B}\sum\nolimits_{b\in\mathcal{B}} v_b\right\|^2 \leq \frac{I(B<n)}{B}\frac{1}{n}\sum_{i=1}^n v_i^2.$$

Furthermore, if the elements in $\mathcal{B}$ are independent, then

$$\mathbb{E}_{\mathcal{B}}\left\|\frac{1}{B}\sum\nolimits_{b\in B} v_b\right\|^2 \leq \frac{1}{Bn}\sum_{i=1}^n v_i^2.$$

Proof. Based on the $\sum_{i=1}^n v_i = \vec{0}$, and permutation and combination, we have

- For the case that $\mathcal{B}$ is a non-empty, uniform random subset of $[n]$, we have

$$\begin{aligned}\mathbb{E}_{\mathcal{B}}\left\|\sum\nolimits_{b\in\mathcal{B}} v_b\right\|^2 =& \mathbb{E}_{\mathcal{B}}\left[\sum\nolimits_{b\in\mathcal{B}}\|v_b\|^2\right] + \frac{1}{C_n^B}\sum\nolimits_{i\in[n]}\left\langle v_i, \frac{C_{n-1}^{B-1}(B-1)}{n-1}\sum\nolimits_{i\neq j} v_j\right\rangle \\ =& B\frac{1}{n}\sum_{i=1}^n v_i^2 + \frac{B(B-1)}{n(n-1)}\sum\nolimits_{i\in[n]}\left\langle v_i, \sum\nolimits_{i\neq j} v_j\right\rangle \end{aligned} \tag{A.2}$$

$$
\begin{aligned}
=&B\frac{1}{n}\sum_{i=1}^{n}v_i^2+\frac{B\left(B-1\right)}{n\left(n-1\right)}\sum\nolimits_{i\in[n]}\left\langle v_i,-v_i\right\rangle\\
=&\frac{B\left(n-B\right)}{\left(n-1\right)}\frac{1}{n}\sum_{i=1}^{n}v_i^2\\
\le&B\mathbb{I}\left(B<n\right)\frac{1}{n}\sum_{i=1}^{n}v_i^2.
\end{aligned}\tag{A.3}
$$

- For the case that the elements in $\mathcal{B}$ are independent, we have

$$
\begin{aligned}
\mathbb{E}_{\mathcal{B}}\left\|\sum\nolimits_{b\in\mathcal{B}}v_b\right\|^2=&\mathbb{E}_{\mathcal{B}}\left[\sum\nolimits_{b\in B}\left\|v_b\right\|^2\right]+2\mathbb{E}_{\mathcal{B}}\left[\sum\nolimits_{1\le b<B}\left\langle v_b,\sum\nolimits_{b<k\le B}v_k\right\rangle\right]\\
=&B\frac{1}{n}\sum\nolimits_{i=1}^{n}\left\|v_i\right\|^2+2\mathbb{E}_{\mathcal{B}}\left[\sum\nolimits_{1\le b<B}\left\langle \mathbb{E}\left[v\right],\sum\nolimits_{b<k\le B}v_k\right\rangle\right]\\
=&B\frac{1}{n}\sum\nolimits_{i=1}^{n}\left\|v_i\right\|^2+B\left(B-1\right)\left\|\mathbb{E}\left[v\right]\right\|^2\\
=&B\frac{1}{n}\sum\nolimits_{i=1}^{n}\left\|v_i\right\|^2.
\end{aligned}\tag{A.4}
$$

$\square$

Lemma A.5. *Consider that $\mathcal{B}$ is a non-empty, uniform random subset of [n] with $|\mathcal{B}|=B$, and the set $\mathcal{D}$ with $|\mathcal{D}|=D$, if $\mathcal{D}$ is a non-empty set, in which each element in $\mathcal{D}$ is* ***independent****, and $\frac{1}{n}\sum_i\mathbb{E}_u\left[h\left(\xi_i,u\right)\right]=\vec{0}$, then*

$$
\mathbb{E}_{\mathcal{B},\mathcal{D}}\left[\left\|\frac{1}{BD}\sum\nolimits_{b\in\mathcal{B}}\sum\nolimits_{j\in\mathcal{D}}h\left(\xi_b,u_j\right)\right\|^2\right]\le\left(\frac{1}{D}+\frac{\mathbb{I}(B<n)}{B}\right)\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_u\|h\left(\xi_i,u\right)\|^2.\tag{A.5}
$$

Proof. $\mathcal{D}$ is a non-empty set, in which each element in $\mathcal{D}$ is **independent**. Consider the $\mathcal{B}$ as an element, and based on the result in Lemma A.4, we have

$$
\begin{aligned}
\left\|\sum\nolimits_{j\in\mathcal{D}}\sum\nolimits_{b\in\mathcal{B}}h\left(\xi_b,u_j\right)\right\|^2=&\sum\nolimits_{j\in\mathcal{D}}\left\|\sum\nolimits_{b\in\mathcal{B}}h\left(\xi_b,u_j\right)\right\|^2\\
&+2\sum\nolimits_{1\le j<D}\left\langle\sum\nolimits_{b\in\mathcal{B}}h\left(\xi_b,u_j\right),\sum\nolimits_{j\le k\le D}\sum\nolimits_{b\in\mathcal{B}}h\left(\xi_b,u_k\right)\right\rangle.
\end{aligned}
$$

Take the expectation with respect to $\mathcal{B}$ and $\mathcal{D}$ for the last two terms, we have

- For the first term,

$$
\begin{aligned}
\mathbb{E}_{\mathcal{B},\mathcal{D}}\left[\sum\nolimits_{j\in\mathcal{D}}\left\|\sum\nolimits_{b\in\mathcal{B}}h\left(\xi_b,u_j\right)\right\|^2\right]\le&\mathbb{E}_{\mathcal{B},\mathcal{D}}\left[\sum\nolimits_{j\in\mathcal{D}}B\sum\nolimits_{b\in\mathcal{B}}\left\|h\left(\xi_b,u_j\right)\right\|^2\right]\\
=&BD\mathbb{E}_{\mathcal{B}}\left[\sum\nolimits_{b\in\mathcal{B}}\mathbb{E}_u\|h\left(\xi_b,u\right)\|^2\right]\\
=&BD\frac{B}{n}\sum_{i=1}^{n}\mathbb{E}_u\|h\left(\xi_i,u\right)\|^2,
\end{aligned}\tag{A.6}
$$

  where (A.6) is based on the fact that $\left(\sum_{i=1}^{n}a_i\right)^2\le n\sum_{i=1}^{n}a_i^2$.

- For the second term,

$$
\begin{aligned}
&2\mathbb{E}_{\mathcal{B},\mathcal{D}}\left[\sum\nolimits_{1\le j<D}\left\langle\sum\nolimits_{b\in\mathcal{B}}h\left(\xi_b,u_j\right),\sum\nolimits_{j<k\le D}\sum\nolimits_{b\in\mathcal{B}}h\left(\xi_b,u_k\right)\right\rangle\right]\\
=&D\left(D-1\right)\mathbb{E}_{\mathcal{B}}\left\|\sum\nolimits_{b\in\mathcal{B}}\mathbb{E}_u\left[h\left(\xi_b,u\right)\right]\right\|^2
\end{aligned}\tag{A.7}
$$

$$=D\left(D-1\right)B\frac{n-B}{n-1}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_u\|h\left(\xi_i,u\right)\|^2, \tag{A.8}$$

where (A.7) follows from independent between j and k, and based on (A.4), and (A.8) follows from (A.3) in Lemma A.4 and the fact $\frac{1}{n}\sum_i\mathbb{E}_u\left[h\left(\xi_i,u\right)\right]=\vec{0}$.

Thus, we have the expectation with respect to $\mathcal{B}$ and $\mathcal{D}$,

$$\begin{aligned}\mathbb{E}_{\mathcal{B},\mathcal{D}}\left\|\sum\nolimits_{b\in\mathcal{B}}\sum\nolimits_{j\in\mathcal{D}}h\left(\xi_b,u_j\right)\right\|^2 \leq & BD\left(B+\left(D-1\right)\frac{n-B}{n-1}\right)\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_u\|h\left(\xi_i,u\right)\|^2\\ \leq & BD(B+D\mathbb{I}(B<n))\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_u\|h\left(\xi_i,u\right)\|^2.\end{aligned}$$

□

A.1. The model of Convergence analysis

Before give the official proof, we give a simple model of convergence sequence, which is easily comprehensive. First, given two sequences,

$$\begin{aligned}\left\|x_{k+1}-\tilde{x}\right\|^2 \leq & a\left\|x_k-\tilde{x}\right\|^2+b\left\|\nabla f\left(x_k\right)\right\|^2;\\ f\left(x_{k+1}\right)\leq & f\left(x_k\right)-p\left\|\nabla f\left(x_k\right)\right\|^2+q\left\|x_k-\tilde{x}\right\|^2.\end{aligned}$$

Define $c_k=(q+c_{k+1}a)$, we can see that

$$\begin{aligned}f\left(x_{k+1}\right)+c_{k+1}\left\|x_{k+1}-\tilde{x}\right\|^2 \leq & f\left(x_k\right)+\left(q+c_{k+1}a\right)\left\|x_k-\tilde{x}\right\|^2-\left(p-bc_{k+1}\right)\left\|\nabla f\left(x_k\right)\right\|^2\\ = & f\left(x_k\right)+c_k\left\|x_k-\tilde{x}\right\|^2-\left(p-bc_{k+1}\right)\left\|\nabla f\left(x_k\right)\right\|^2\end{aligned}$$

if parameters $a,b,p,q>0$ satisfy, $\forall k>0$,

- $c_0>0$ and $c_k\geq c_{k+1}>0$ is a decrease sequence;
- $p-bc_{k+1}\geq p-bc_0>0$.

Thus, we can obtain

$$\frac{1}{K}\sum_{k=0}^{K-1}\left\|\nabla f\left(x_k\right)\right\|_2^2\leq\frac{f\left(x_0\right)+c_0\left\|x_0-\tilde{x}\right\|_2^2-\left(f\left(x_K\right)+c_K\left\|x_K-\tilde{x}\right\|_2^2\right)}{K\left(p-bc_0\right)}.$$

How to choose the parameters and how to compute the best complexity of iteration or the gradient will be based on the algorithm we proposed and the property of the function we use.

B. Convergence proof for Smooth function with Gaussian smooth

B.1. Algorithm: mini-batch SZVR-G

We present our mini-batch SZVR-G here in Algorithm 2.

B.2. Convergence tool

In this section, we focus on Algorithm 1 that apply to Gaussian-smoothed function, and mainly give the upper bounds for $\mathbb{E}_u\|G_\mu(\tilde{x}_s,u_\mathcal{D},\xi_\mathcal{B})-\nabla f_\mu(\tilde{x}_s)\|^2$, $\mathbb{E}_{i,j,u}\|G_\mu(x_k,u_{\mathcal{D}[j]},\xi_i)-G_\mu(\tilde{x}_s,u_{\mathcal{D}[j]},\xi_i)\|^2$ and $\mathbb{E}_{i,j,u}\|\tilde{\nabla}_k\|^2$, which are used for analyzing the convergence sequence. Note, we drop the superscript i and k of ξ and u, respectively, for focusing on a single epoch analysis.

Algorithm 2 Mini-batch Zeroth-order via variance reduction with Gaussian smooth

Require: K, S, η (learning rate), and $\tilde{x}_0$

for $s = 0, 1, 2, \cdots, S-1$ **do**

  Independently Generate Gaussian vector set $u_{\mathcal{D}}$ through Gaussian random vector generator with D times, where $\mathcal{D}$ is the index set. ▷ In practice, store Gaussian random vector seeds for each sth iteration.

  Sample from $[n]$ to form mini-batch $\mathcal{B}$ with $|\mathcal{B}| = B$.

  $x_0 = \tilde{x}_s$

  $G = G_\mu(x, u_{\mathcal{D}}, \xi_{\mathcal{B}})$ ▷ (3.1)

  for $k = 0, 1, 2, \cdots, K-1$ **do**

    $\Lambda = 0$

    for $t = 0, 1, 2, \cdots, b_0 - 1$ **do**

      Sample i from $[n]$ and j from $[D]$

      $\Lambda_{t+1} = \Lambda_t + G_\mu\left(x_k, u_{\mathcal{D}[j]}, \xi_i\right) - G_\mu\left(\tilde{x}_s, u_{\mathcal{D}[j]}, \xi_i\right)$

    end for

    $\tilde{\nabla}_k = \Lambda_{b_0}/b_0 + G$

    $x_{k+1} = x_k - \eta \tilde{\nabla}_k$

  end for

  Update $\tilde{x}_{s+1} = x_K$

end for

Output: $\tilde{x}_k^s$, $s \in \{1, ..., S\}$, $k \in \{1, ..., K\}$

Lemma B.1. *In Algorithm 1, for $F(x,\xi) \in C^{1,1}$ and $G_\mu\left(\tilde{x}_s, u_{\mathcal{D}}, \xi_{\mathcal{B}}\right)$ defined in (3.1), we have*

$$\mathbb{E}_{u,\mathcal{B}} \|G_\mu\left(\tilde{x}_s, u_{\mathcal{D}}, \xi_{\mathcal{B}}\right) - \nabla f_\mu\left(\tilde{x}_s\right)\|^2 \leq \left(\frac{1}{D} + \frac{\mathbb{I}(B<n)}{B}\right)\left(\frac{\mu^2}{2} L_1^2 (d+6)^3 + 2\left(d+4\right)\|\nabla f\left(x\right)\|^2\right).$$

Proof. By the definition of $G_\mu\left(\tilde{x}_s, u_{\mathcal{D}}, \xi_{\mathcal{B}}\right)$ defined in (3.1), we have

$$\begin{aligned}\mathbb{E}_{u,\mathcal{B}} \|G_\mu\left(\tilde{x}_s, u_{\mathcal{D}}, \xi_{\mathcal{B}}\right) - \nabla f_\mu\left(\tilde{x}_s\right)\|^2 &= \left(\frac{1}{D} + \frac{\mathbb{I}(B<n)}{B}\right)\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_u \|G_\mu\left(x, u, \xi_i\right) - \nabla f_\mu\left(\tilde{x}_s\right)\|^2 &\text{(B.1)}\\ &\leq \left(\frac{1}{D} + \frac{\mathbb{I}(B<n)}{B}\right)\left(\frac{\mu^2}{2} L_1^2 (d+6)^3 + 2\left(d+4\right)\|\nabla f\left(x\right)\|\right), &\text{(B.2)}\end{aligned}$$

where (B.1) based on Lemma A.5 that the vector in $u_{\mathcal{D}}$ is independent, (B.2) is based on the Lemma 2.1. □

The following Lemma can be obtained directly from Lemma A.4 under the requirement that $\frac{1}{n}\sum_{i=1}^{n}\left(\nabla f_\mu(x) - \nabla F_\mu(x,\xi_i)\right) = 0$, and Assumption 2.1.

Lemma B.2. *In Algorithm 1, for $F(x,\xi) \in C^{1,1}$ and $\nabla f_\mu\left(x\right)$ defined in (2.1), we have*

$$\frac{1}{B}\sum\nolimits_{i=1}^{B} \left\|\nabla f_\mu\left(x\right) - \nabla F_\mu\left(x, \xi_{\mathcal{B}[i]}\right)\right\|^2 \leq \frac{\mathbb{I}(B<n)}{B} H.$$

Lemma B.3. *In Algorithm 1, for $F(x,\xi) \in C^{1,1}$, $\mu > 0$, and $G_\mu\left(x, u, \xi\right)$ defined in (1.2), we have*

$$\mathbb{E}_{i,j,u} \left\|G_\mu\left(x_k, u_{D[j]}, \xi_i\right) - G_\mu\left(\tilde{x}_s, u_{D[j]}, \xi_i\right)\right\|^2 \leq \frac{3}{2} L_1^2 \mu^2 K (d+6)^3 + 3L_1^2 (d+4)\|x_k - \tilde{x}_s\|^2.$$

Proof. We first drop the subscript of i and $\mathcal{D}\left[j\right]$ in $\left\|G_\mu\left(x_k, u_{\mathcal{D}[j]}, \xi_i\right) - G_\mu\left(\tilde{x}_s, u_{\mathcal{D}[j]}, \xi_i\right)\right\|$ for simple and easily understanding. Through adding and subtracting the terms $-\mu\left\langle \nabla F\left(\tilde{x}_s, \xi\right), u\right\rangle + \mu\left\langle \nabla F\left(\tilde{x}_s, \xi\right), u\right\rangle$, and by the definition of $G_\mu\left(x_k, \xi, u\right)$ in 1.2, we have

$$\begin{aligned}&\|G_\mu\left(x_k, \xi, u\right) - G_\mu\left(\tilde{x}_s, \xi, u\right)\|^2\\ =&\left\|\frac{F\left(x_k + \mu u, \xi\right) - F\left(x_k, \xi\right)}{\mu} u - \frac{F\left(\tilde{x}_s + \mu u, \xi\right) - F\left(\tilde{x}_s, \xi\right)}{\mu} u\right\|^2\end{aligned}$$

$$
\begin{aligned}
=&\frac{\|u\|^2}{\mu^2}\left(F\left(x_k+\mu u,\xi\right)-F\left(x_k,\xi\right)-\left(F\left(\tilde{x}_s+\mu u,\xi\right)-F\left(\tilde{x}_s,\xi\right)\right)\right)^2\\
=&\frac{\|u\|^2}{\mu^2}(F\left(x_k+\mu u,\xi\right)-F\left(x_k,\xi\right)-\mu\left\langle\nabla F\left(x_k,\xi\right),u\right\rangle-\left(F\left(\tilde{x}_s+\mu u,\xi\right)-F\left(\tilde{x}_s,\xi\right)-\mu\left\langle\nabla F\left(\tilde{x}_s,\xi\right),u\right\rangle\right)\\
&+\mu\left\langle\nabla F\left(x_k,\xi\right),u\right\rangle-\mu\left\langle\nabla F\left(\tilde{x}_s,\xi\right),u\right\rangle)^2\\
\leq&3\frac{\|u\|^2}{\mu^2}\left(F\left(x_k+\mu u,\xi\right)-F\left(x_k,\xi\right)-\mu\left\langle\nabla F\left(x_k,\xi\right),u\right\rangle\right)^2\\
&+3\frac{\|u\|^2}{\mu^2}\left(F\left(\tilde{x}_s+\mu u,\xi\right)-F\left(\tilde{x}_s,\xi\right)-\mu\left\langle\nabla F\left(\tilde{x}_s,\xi\right),u\right\rangle\right)^2\\
&+3\frac{\|u\|^2}{\mu^2}\left(\mu\left\langle\nabla F\left(x_k,\xi\right),u\right\rangle-\mu\left\langle\nabla F\left(\tilde{x}_s,\xi\right),u\right\rangle\right)^2\\
\leq&3\frac{\|u\|^2}{\mu^2}\left(\left(\frac{L_1\mu^2}{2}\|u\|^2\right)^2+\left(\frac{L_1\mu^2}{2}\|u\|^2\right)^2+\left\langle\nabla F\left(x_k,\xi\right)-\nabla F\left(\tilde{x}_s,\xi\right),u\right\rangle^2\right)\\
\leq&\frac{3}{2}L_1^2\mu^2\|u\|^6+3\langle\nabla F\left(x_k,\xi\right)-\nabla F\left(\tilde{x}_s,\xi\right),u\rangle^2\|u\|^2,
\end{aligned}
$$

where the first inequality follows from $(a_1+a_2+a_3)^2\leq 3a_1^2+3a_2^2+3a_3^2$, the second inequality is based on the smoothness of $F(x,\xi)$ and $\langle b_1,b_2\rangle\leq\|b_1\|\,\|b_2\|$. the last inequality follows from smoothness of $F(x,\xi)$. Take expectation with respect to j, i and u, we have

$$
\begin{aligned}
&E_{i,j,u}\left\|G_\mu\left(x_k,u_{D[j]},\xi_i\right)-G_\mu\left(\tilde{x}_s,u_{D[j]},\xi_i\right)\right\|^2\\
\leq&E_{i,j,u}\left[\frac{3}{2}L_1^2\mu^2\|u\|^6+3\mu^2(\langle\nabla F\left(x_k,\xi\right)-\nabla F\left(\tilde{x}_s,\xi\right),u\rangle)^2\right]\\
\leq&\frac{3}{2}L_1^2\mu^2(d+6)^3+3\left(d+4\right)\|\nabla F\left(x_k,\xi\right)-\nabla F\left(\tilde{x}_s,\xi\right)\|^2,\\
\leq&\frac{3}{2}L_1^2\mu^2(d+6)^3+3L_1^2\left(d+4\right)\|x_k-\tilde{x}_s\|^2,
\end{aligned}
$$

where the second inequality is based on Lemma A.2 for $p=6$ and $p=4$ and **Lemma A.3, which is an important lemma to strengthen the upper bound**; The last inequality is based on the smoothness of F. ∎

Lemma B.4. *In Algorithm 1, for $F(x,\xi)\in C^{1,1}$, $\mu>0$, and $\tilde{\nabla}_k$ defined in (3.2), we have*

$$
\begin{aligned}
\mathbb{E}_{i,j,u,\mathcal{B}}\|\tilde{\nabla}_k\|^2\leq&\frac{9}{2}L_1^2\mu^2(d+6)^3+9L_1^2(d+4)\|x_k-\tilde{x}_s\|^2+3\|\nabla f_\mu\left(x_k\right)\|^2\\
&+3\left(\frac{1}{D}+\frac{\mathbb{I}(B<n)}{B}\right)\left(\frac{\mu^2}{2}L_1^2(d+6)^3+2\left(d+4\right)\|\nabla f\left(x\right)\|^2\right).
\end{aligned}
$$

Proof. By adding and subtracting terms $\nabla f_\mu\left(x_k\right)$ and $\nabla f_\mu\left(\tilde{x}_s\right)$, we have,

$$
\begin{aligned}
\|\tilde{\nabla}_k\|^2=&||G_\mu\left(x_k,u_{\mathcal{D}[j]},\xi_i\right)-G_\mu\left(\tilde{x}_s,u_{\mathcal{D}[j]},\xi_i\right)+G_\mu\left(\tilde{x}_s,u_{\mathcal{D}},\xi_{\mathcal{B}}\right)\\
&-\nabla f_\mu\left(x_k\right)+\nabla f_\mu\left(\tilde{x}_s\right)+\nabla f_\mu\left(x_k\right)-\nabla f_\mu\left(\tilde{x}_s\right)||^2\\
\leq&3\left\|G_\mu\left(x_k,u_{D[j]},\xi_i\right)-G_\mu\left(\tilde{x}_s,u_{D[j]},\xi_i\right)-\left(\nabla f_\mu\left(x_k\right)-\nabla f_\mu\left(\tilde{x}_s\right)\right)\right\|^2\\
&+3\|\nabla f_\mu\left(x_k\right)\|^2+3\|G_\mu\left(\tilde{x}_s,u_{\mathcal{D}},\xi_{\mathcal{B}}\right)-\nabla f_\mu\left(\tilde{x}_s\right)\|^2,
\end{aligned}
\tag{B.3}
$$

where (B.3) is based on the fact that $(a_1+a_2+a_3)^2\leq 3a_1^2+3a_2^2+3a_3^2$. Taking expectation with respect to i, j and u, we have

$$
\begin{aligned}
\mathbb{E}_{i,j,u,\mathcal{B}}\left\|\tilde{\nabla}_k\right\|^2\leq&3\mathbb{E}_{i,j,u}\left\|G_\mu\left(x_k,u_{\mathcal{D}[j]},\xi_i\right)-G_\mu\left(\tilde{x}_s,u_{\mathcal{D}[j]},\xi_i\right)-\left(\nabla f_\mu\left(x_k\right)-\nabla f_\mu\left(\tilde{x}_s\right)\right)\right\|^2\\
&+3\|\nabla f_\mu\left(x_k\right)\|^2+3\mathbb{E}_{u,\mathcal{B}}\|G_\mu\left(\tilde{x}_s,u_{\mathcal{D}},\xi_{\mathcal{B}}\right)-\nabla f_\mu\left(\tilde{x}_s\right)\|^2
\end{aligned}
$$

$$\leq 3\mathbb{E}_{i,j,u}\left\|G_\mu\left(x_k, u_{\mathcal{D}[j]}, \xi_i\right) - G_\mu\left(\tilde{x}_s, u_{\mathcal{D}[j]}, \xi_i\right)\right\|^2 + 3\|\nabla f_\mu\left(x_k\right)\|^2 \tag{B.4}$$
$$+ 3\mathbb{E}_{u,\mathcal{B}}\|G_\mu\left(\tilde{x}_s, u_{\mathcal{D}}, \xi_{\mathcal{B}}\right) - \nabla f_\mu\left(\tilde{x}_s\right)\|^2$$
$$\leq \frac{9}{2}L_1^2\mu^2(d+6)^3 + 9L_1^2(d+4)\|x_k - \tilde{x}_s\|^2 + 3\|\nabla f_\mu\left(x_k\right)\|^2 \tag{B.5}$$
$$+ 3\left(\frac{1}{D} + \frac{\mathbb{I}(B<n)}{B}\right)\left(\frac{\mu^2}{2}L_1^2(d+6)^3 + 2\left(d+4\right)\|\nabla f\left(x\right)\|^2\right), \tag{B.6}$$

where (B.4) the inequality follows from $\mathbb{E}\|Z - EZ\|^2 \leq \mathbb{E}\|Z\|^2$, where Z is a random variable; (B.5) and (B.6) are based on Lemma B.3 and Lemma B.1. Note that, for convenience, we further take expectation with respect to $\mathcal{B}$ in the last equality. ☐

B.3. Convergence analysis

In this subsection, mainly based on Lemma B.4, smoothness and update of x in Algorithm 1, we give the new sequence of the proposed algorithm: $\mathbb{E}_{i,j}\left[f_\mu\left(x_{k+1}\right)\right] + c_{k+1}\mathbb{E}_{i,j}\|x_{k+1} - \tilde{x}_s\|^2$. In order to obtain the convergence sequence,we provide the formulation of the sequence c_k, w_k and J_k, which is the key parameter in analyzing the convergence and complexity. In Remark B.1 and B.2, we analyze the the parameter's relationship between K, q and η such that these new formed sequence can be converged.

Lemma B.5. *In Algorithm 1, for $F(x,\xi) \in C^{1,1}$, $\mu > 0$, $q > 0$, we have*

$$\frac{1}{S}\sum_{s=1}^{S}\frac{1}{K}\sum_{k=0}^{K-1} w_k\|\nabla f_\mu\left(x_k^s\right)\|^2 - \frac{1}{SK}\sum_{s=0}^{S-1}\sum_{k=0}^{K-1}\beta_k\|\nabla f\left(x\right)\|^2 \leq \frac{R}{SK} + J_{k+1},$$

where x^* is the optimal value of function $f_\mu(x)$, $R = \max_x\{f_\mu(x) - f_\mu(x_*) : f_\mu(x) \leq f_\mu(x_0)\}$, and

$$c_k = \left(1 + q\eta + 9L_1^2\left(d+4\right)\eta^2\right)c_{k+1} + \frac{9}{2}L_1^3\left(d+4\right)\eta^2 \tag{B.7}$$
$$J_k = \frac{3}{2}\left(\frac{1}{D} + \frac{\mathbb{I}(B<n)}{B} + 3\right)\left(L_1 + 2c_{k+1}\right)\mu^2 L_1^2(d+6)^3\eta^2$$
$$+ \left(1 + \frac{1}{q}c_{k+1}\right)\frac{1}{2}\eta\frac{\mathbb{I}(B<n)}{B}H, \tag{B.8}$$
$$w_k = \left(\frac{1}{2} - \frac{2}{q}c_{k+1}\right)\eta - 3\left(\frac{L_1}{2} + c_{k+1}\right)\eta^2, \tag{B.9}$$
$$\beta_k = 6\left(\frac{1}{D} + \frac{\mathbb{I}(B<n)}{B}\right)\left(d+4\right)\left(\frac{L_1}{2} + c_{k+1}\right)\eta^2 \tag{B.10}$$

Proof. In each inner iteration, we use x_k^s to indicate the iteration, but we drop the s for simple. Based on the smoothness of $f_\mu(x)$ and update of x_k in Algorithm 1, take expectation with respect to i, j and u and $\mathbb{E}_{i,j,u}[\tilde{\nabla}_k]$ in 3.3, we have,

$$\begin{aligned}
&E_{i,j,u}\left[f_\mu\left(x_{k+1}\right)\right] \\
\leq & f_\mu(x_k) - \eta\langle\nabla f_\mu(x_k), E_{i,j,u}[\tilde{\nabla}_k]\rangle + \frac{L_1\eta^2}{2}E_{i,j,u}\|\tilde{\nabla}_k\|^2 \\
= & f_\mu\left(x_k\right) - \eta\left\langle\nabla f_\mu\left(x_k\right), \nabla f_\mu\left(x_k\right) - \nabla f_\mu\left(\tilde{x}_s\right) + \nabla F_\mu\left(\tilde{x}_s, \xi_{\mathcal{B}}\right)\right\rangle + \frac{L_1\eta^2}{2}E_{i,j,u}\left\|\tilde{\nabla}_k\right\|^2 \\
\leq & f_\mu\left(x_k\right) - \eta\|\nabla f_\mu\left(x_k\right)\|^2 + \frac{1}{2}\eta\|\nabla f_\mu\left(x_k\right)\|^2 + \frac{1}{2}\eta\left\|\nabla f_\mu\left(\tilde{x}_s\right) + \nabla F_\mu\left(\tilde{x}_s, \xi_{\mathcal{B}}\right)\right\| + \frac{L_1\eta^2}{2}E_{i,j,u}\left\|\tilde{\nabla}_k\right\|^2; \\
&E_{i,j,u}\|x_{k+1} - \tilde{x}_s\|^2 \\
= & E_{i,j,u}\left\|x_k - \eta\tilde{\nabla}_k - \tilde{x}_s\right\|^2 = \|x_k - \tilde{x}_s\|^2 - 2\eta\langle x_k - \tilde{x}_s, E_{i,j}[\tilde{\nabla}_k]\rangle + \eta^2 E_{i,j,u}\|\tilde{\nabla}_k\|^2 \\
= & \|x_k - \tilde{x}_s\|^2 - 2\eta\langle x_k - \tilde{x}_s, \nabla f_\mu\left(x_k\right) - \nabla f_\mu\left(\tilde{x}_s\right) + \nabla F_\mu\left(\tilde{x}_s, \xi_{\mathcal{B}}\right)\rangle + \eta^2 E_{i,j,u}\left\|\tilde{\nabla}_k\right\|^2 \\
\leq & \|x_k - \tilde{x}_s\|^2 + \eta q\|x_k - \tilde{x}_s\|^2 + 2\frac{1}{q}\eta\|\nabla f_\mu\left(x_k\right)\|^2 + 2\frac{1}{q}\eta\|\nabla f_\mu\left(\tilde{x}_s\right) + \nabla F_\mu\left(\tilde{x}_s, \xi_{\mathcal{B}}\right)\|^2 + \eta^2 E_{i,j,u}\left\|\tilde{\nabla}_k\right\|^2,
\end{aligned}$$

where the last equality is based on $2\langle a_1, a_2\rangle \le \frac{1}{q}\|a_1\|^2 + q\|a_2\|^2$. **The key difference between the proof of SVRG [20] is the upper bound that separating the term $\nabla f_\mu(x_k)$ and $-\nabla f_\mu(\tilde{x}_s) + \nabla F_\mu(\tilde{x}_s, \xi_{\mathcal{B}})$, due to the fact that they are dependent with respect to $\mathcal{B}$**. For convenience, we take expectation with respect to $\mathcal{B}$ and apply Lemma B.2, thus, we have

$$
\begin{aligned}
E_{i,j,u}\left[f_\mu(x_{k+1})\right] \le& f_\mu(x_k) - \frac{1}{2}\eta\|\nabla f_\mu(x_k)\|^2 + \frac{1}{2}\eta\frac{\mathbb{I}(B<n)}{B}H + \frac{L_1\eta^2}{2}E_{i,j,u}\|\tilde{\nabla}_k\|^2;\\
E_{i,j,u}\|x_{k+1}-\tilde{x}_s\|^2 \le& \|x_k-\tilde{x}_s\|^2 + \eta q\|x_k-\tilde{x}_s\|^2 + 2\frac{1}{q}\eta\|\nabla f_\mu(x_k)\|^2 + 2\frac{1}{q}\eta\frac{\mathbb{I}(B<n)}{B}H + \eta^2 E_{i,j,u}\|\tilde{\nabla}_k\|^2
\end{aligned}
$$

Plus $\mathbb{E}_{i,j,u}\left[f_\mu(x_{k+1})\right]$ with $c_{k+1}\mathbb{E}_{i,j,u}\|x_{k+1}-\tilde{x}_s\|^2$, and apply Lemma B.4, we have

$$
\begin{aligned}
&\mathbb{E}_{i,j,u}\left[f_\mu(x_{k+1})\right] + c_{k+1}\mathbb{E}_{i,j,u}\|x_{k+1}-\tilde{x}_s\|^2\\
\ge& f_\mu(x_k) + c_k\|x_k-\tilde{x}_s\|^2 - w_k\|f_\mu(x_k)\|^2 + J_k + \beta_k\|\nabla f(x_k)\|^2,
\end{aligned}
$$

where

$$
\begin{aligned}
c_k =& \left(1+q\eta+9L_1^2(d+4)\eta^2\right)c_{k+1} + \frac{9}{2}L_1^3(d+4)\eta^2\\
w_k =& \left(\frac{1}{2}-\frac{2}{q}c_{k+1}\right)\eta - 3\left(\frac{L_1}{2}+c_{k+1}\right)\eta^2,\\
J_k =& \frac{3}{2}\left(\frac{1}{D}+\frac{\mathbb{I}(B<n)}{B}+3\right)(L_1+2c_{k+1})\mu^2L_1^2(d+6)^3\eta^2 + \left(1+\frac{1}{q}c_{k+1}\right)\frac{1}{2}\eta\frac{\mathbb{I}(B<n)}{B}H,\\
\beta_k =& 6\left(\frac{1}{D}+\frac{\mathbb{I}(B<n)}{B}\right)(d+4)\left(\frac{L_1}{2}+c_{k+1}\right)\eta^2.
\end{aligned}
$$

Summing up $\mathbb{E}_{i,j}\left[f_\mu(x_k)\right] + c_k\mathbb{E}_{i,j}\|x_k-\tilde{x}_s\|^2$ from $k=0$ to $k=K-1$, we have

$$
\begin{aligned}
&E_{i,j,u}\left[f_\mu(x_{k+1})\right] + c_{k+1}E_{i,j,u}\|x_{k+1}-\tilde{x}_s\|^2\\
\le& f_\mu(x_k) + c_0\|x_0-\tilde{x}_s\|^2 + KJ_k + \sum_{k=0}^{K-1} -w_k\|\nabla f_\mu(x_k)\|^2 + \sum_{k=0}^{K-1}\beta_k\|\nabla f(x)\|^2.
\end{aligned}
$$

Thus, based on $x_0 = \tilde{x}_s$ and $\tilde{x}_{s+1} = s_K$ we have

$$
\begin{aligned}
&\frac{1}{K}\sum_{k=0}^{K-1} w_k\|\nabla f_\mu(x_k)\|^2 - \sum_{k=0}^{K-1}\beta_k\|\nabla f(x)\|^2\\
\le& \frac{f_\mu(x_k) + c_0\|x_0-\tilde{x}_s\|^2 - \left(E_{i,j,u}\left[f_\mu(x_K)\right] + c_K E_{i,j,u}\|x_K-\tilde{x}_s\|^2\right)}{K} + J_{k+1}\\
\le& \frac{f_\mu(x_k) + c_0\|x_0-\tilde{x}_s\|^2 - E_{i,j,u}\left[f_\mu(x_K)\right]}{K} + J_{k+1}\\
=& \frac{f_\mu(x_s) - E_{i,j,u}\left[f_\mu(\tilde{x}_{s+1})\right]}{K} + J_{k+1}.
\end{aligned}
$$

Summing up from $s=0$ to $s=S-1$, (here we add the s to x_k to indicate different epoch s), thus we have

$$
\begin{aligned}
&\frac{1}{S}\sum_{s=1}^{S}\frac{1}{K}\sum_{k=0}^{K-1} w_k\|\nabla f_\mu(x_k^s)\|^2 - \sum_{k=0}^{K-1}\beta_k\|\nabla f(x)\|^2\\
\le& \frac{f_\mu(x_0) - E_{i,j,u}\left[f_\mu(\tilde{x}_S)\right]}{SK} + J_{k+1}\\
\le& \frac{f_\mu(x_0) - f_\mu(x_*)}{SK} + J_{k+1}\\
\le& \frac{R}{SK} + J_{k+1},
\end{aligned}
$$

where x^* is the optimal value of function $f_\mu(x)$, $R = \max_x\{f_\mu(x) - f_\mu(x_*) : f_\mu(x) \le f_\mu(x_0)\}$,

$\square$

B.3.1 Parameters analysis

In order to satisfy the convergence with the sequence $\{u_k\}$, $\{c_k\}$, and $\{J_k\}$, we consider the parameters setting, such as K, q, η, which will be used to analyze the complexity of $\mathcal{SZO}$.

Remark B.1. *For c_k in (B.7). Due to the fact that,*

$$c_k = \left(1+\eta q + 9L_1^2\left(d+4\right)\eta^2\right)c_{k+1} + \frac{9}{2}L_1^3\left(d+4\right)\eta^2 \geq \left(1+q\eta+9L_1^2(d+4)\eta^2\right)c_{k+1} \geq c_{k+1},$$

c_k is decreasing function, that is $c_k \leq c_0$. Then, we can see that $w_k \geq w_0$, $J_k \leq J_0$. Thus, we only consider c_0, w_0, and J_0. Here, analyze c_0

- *For c_0, based on Lemma A.1, we have*

$$c_k = \left(\frac{1}{Y}\right)^k\left(c_0 + \frac{U}{Y-1}\right) - \frac{U}{Y-1},$$

where $Y = 1+q\eta+9L_1^2(d+4)^2\eta^2$, and $U = \frac{9}{2}L_1^3(d+4)^2\eta^2$. In order to ensure $c_k > 0$, and there exists a large $K > 0$ and a lower bound for c_0, that is,

$$c_0 \leq \frac{U\left(Y^K-1\right)}{Y-1} = \frac{\frac{9}{2}L_1^3(d+4)\eta^2}{q\eta+9L_1^2(d+4)\eta^2}C \leq \frac{1}{2}L_1 C, \tag{B.11}$$

where $C = \left(1+q\eta+9L_1^2(d+4)\eta^2\right)^K - 1$.

- *For $C = (1+q\eta+9L_1^2(d+4)\eta^2)^K - 1$ defined above. In order to keep the upper bound of $c_0 \nrightarrow +\infty$, we use the function with the special structure and characteristic of $(1+t^1)^{t^2} \to e^2$, where $t^1t^2 \leq 1, 0 < t^1 < 1$. Thus, we consider the number of inner iteration K that has the relationship with q and η, that is*

$$(q\eta+9L_1^2(d+4)\eta^2)K \leq 1, q\eta+9L_1^2(d+4)\eta^2 < 1. \tag{B.12}$$

Then, we can see that $C \leq e-1 \approx 1.7$, and $c_0 \leq L_1$.

Remark B.2. *For $w_0 = (1/2 - 2c_0/q)\eta - (3L_1/2+3c_0)\eta^2$ defined in (B.9), should be positive. In order to have small η such that $w_0 = O(\eta)$, we should require $(1/2-2c_0/q)$ is positive, that is $c_0/q \leq O(1)$. We consider c_0/q, and the relationship with w_0 and η, separately,*

- *For c_0/q. we set $c_0/q \leq 1/4$ for convenience. Specifically, $C \leq e-1 \approx 1.7$.*

$$\frac{1}{q}c_0 = \frac{1}{q}\frac{\frac{9}{2}L_1^3(d+4)\eta^2}{q\eta+9L_1^2(d+4)\eta^2}C \leq \frac{1}{q}\frac{\frac{9}{2}L_1^3(d+4)\eta}{q+9L_1^2(d+4)\eta} \leq \frac{1}{4}. \tag{B.13}$$

By arrangement, we require the function with respect to q,

$$q^2 + 9L^2(d+4)\eta q - 36L^3(d+4)\eta \geq 0,$$

Based on $\sqrt{a_1^2+a_2^2} \leq a_1+a_2$, $(a_1,a_2 \geq 0)$, the solution of above quadratic square, q_2, is

$$\begin{aligned} q_2 =& \frac{-9L_1^2(d+4)\eta + \sqrt{(9L_1^2(d+4)\eta)+12L_1^3(d+4)\eta}}{2} \\ \leq& \frac{-9L_1^2(d+4)\eta + 9L_1^2(d+4)\eta + \sqrt{144L_1^3(d+4)\eta}}{2} = 6\sqrt{L_1^3(d+4)\eta}. \end{aligned}$$

We set

$$q = 6\sqrt{L_1^3(d+4)\eta}, \tag{B.14}$$

that satisfy (B.13). Based on the setting of q, we have

[2] Here'e' is the Euler number, 2.718

- *Consider the requirement in (B.12) that $q\eta + 9L_1^2(d+4)\eta^2 < 1$, we require that*

$$\eta \leq \frac{1}{9d^{1/2}L_1} < O\left(\frac{1}{d^{1/3}}\right). \tag{B.15}$$

- *Furthermore, the inequality in (B.12) that $(q\eta + 9L_1^2(d+4)\eta^2)K \leq 1$, and q in (B.14), we require that*

$$K \leq \frac{2}{\max\{d\eta^2, q\eta\}} = \frac{2}{\max\{d\eta^2, d^{0.5}\eta^{1.5}\}} \tag{B.16}$$

- *For w_0 and η, with convenience, we set*

$$w_0 = \left(\frac{1}{2} - \frac{1}{q}c_0\right)\eta - (L_1 + 2c_0)\,\eta^2 \geq \frac{1}{4}\eta - (L_1 + 2c_0)\,\eta^2 \geq \frac{1}{8}\eta, \tag{B.17}$$

where the first inequality is based on $\frac{1}{q}c_0 \leq \frac{1}{4} \Rightarrow \frac{1}{2} - \frac{1}{q}c_0 \geq \frac{1}{2}$. In order to obtain $w_0 \geq \frac{1}{8}\eta$, using $c_0 \leq L_1$ in (B.11), as long as

$$\eta \leq \frac{1}{24L_1} \leq \frac{1}{8(L_1 + 2L_1)} \overset{c_0 \leq L_1}{\leq} \frac{1}{8(L_1 + 2c_0)}.$$

However, combing with the range of η in (B.15), we can see that the above equality is success immediately.

B.3.2 Convergence of $\|\nabla f(x)\|^2$

Based on the above setting analysis: (B.12), (B.14), (B.16) and (B.17), we reconsider the new sequence. Moreover, based on the relationship of $\|\nabla f(x)\|^2$ and $\|\nabla f_\mu(x)\|^2$ in Lemma 2.1 and the convergence of $\|\nabla f_\mu(x)\|^2$, we present the convergence of $\|\nabla f(x)\|^2$.

Theorem. *3.1 In Algorithm 1, under Assumption 2.1, for $F(x,\xi) \in C^{1,1}$, let parameters $\mu, \eta, q, K > 0$, $c_0 \leq L_1$ and the cardinality of Gaussian vector set and sample set $D \geq O(\eta d)$ and $B \geq O(\min\{\eta d, n\})$, we have*

$$\frac{1}{S}\sum_{s=1}^{S}\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla f(x_k^s)\|^2 \leq \frac{32R}{SK\eta} + \frac{32}{\eta}J_0 + \frac{1}{2}\mu^2 L_1^2(d+6)^3,$$

where x^* is the optimal value of function $f_\mu(x)$, $R = \max_x\{f_\mu(x) - f_\mu(x_*) : f_\mu(x) \leq f_\mu(x_0)\}$, and

$$J_0 = \frac{3}{4}\left(\frac{1}{D} + \frac{\mathbb{I}(B<n)}{B} + 3\right)(L_1 + 2c_0)\,\mu L_1^2(d+6)^3\eta^2 + \left(1 + \frac{1}{q}c_0\right)\frac{1}{2}\eta\frac{\mathbb{I}(B<n)}{B}H. \tag{B.18}$$

Proof. Base on the parameters that c_k is decreasing sequence and $c_0 > c_k$, $w_0 < w_k$, and $\beta_k < \beta_0$, $J_k < J_0$, following Lemma B.5, we have

$$\begin{aligned}
&\frac{1}{S}\sum_{s=1}^{S}\frac{1}{K}\sum_{k=0}^{K-1}w_0\|\nabla f_\mu(x_k^s)\|^2 - \frac{1}{SK}\sum_{s=0}^{S-1}\sum_{k=0}^{K-1}\beta_0\|\nabla f(x)\|^2\\
\leq&\frac{1}{S}\sum_{s=1}^{S}\frac{1}{K}\sum_{k=0}^{K-1}w_k\|\nabla f_\mu(x_k^s)\|^2 - \frac{1}{SK}\sum_{s=0}^{S-1}\sum_{k=0}^{K-1}\beta_k\|\nabla f(x)\|^2\\
\leq&\frac{R}{SK} + J_{k+1}\\
\leq&\frac{R}{SK} + J_0,
\end{aligned}$$

Combing with the inequality in Lemma 2.1 that replace the smoothed $\nabla f_\mu(x)$ with $\nabla f(x)$, we have

$$\left(\frac{1}{2}w_0 - \beta_0\right)\frac{1}{SK}\sum_{s=1}^{S}\sum_{k=0}^{K-1}\left\|\nabla f(x_s^k)\right\|^2 - \frac{1}{2}w_0\mu^2 L_1^2(d+6)^3$$

$$
\begin{aligned}
=&\frac{1}{SK}\sum_{s=1}^{S}\sum_{k=0}^{K-1}\frac{1}{2}w_0\left(\left\|\nabla f\left(x_s^k\right)\right\|^2-\frac{1}{2}\mu^2L_1^2(d+6)^3\right)-\beta_0\frac{1}{SK}\sum_{s=0}^{S-1}\sum_{k=0}^{K-1}\left\|\nabla f\left(x_s^k\right)\right\|^2\\
=&w_0\frac{1}{SK}\sum_{s=0}^{S-1}\sum_{k=0}^{K-1}\left\|\nabla f\left(x_s^k\right)\right\|^2-\beta_0\frac{1}{SK}\sum_{s=0}^{S-1}\sum_{k=0}^{K-1}\left\|\nabla f\left(x_s^k\right)\right\|^2\\
\leq&\frac{R}{SK}+J_0,
\end{aligned}
$$

In order to keep $\frac{1}{2}w_0-\beta_0>0$, based on $w_0\geq\frac{1}{8}\eta$ in (B.17), we require $\beta_0\leq\eta/32$, based on the definition of β_k in (B.10), we have,

$$
6\left(\frac{1}{D}+\frac{\mathbb{I}(B<n)}{B}\right)(d+4)\left(\frac{L_1}{2}+c_{k+1}\right)\eta^2\leq\frac{1}{32}\eta
$$

thus, we require

$$
D\geq O\left(\eta d\right),B\geq O\left(\min\{\eta d,n\}\right). \tag{B.19}
$$

Base on the parameters setting in (B.17) that replace w_0 with $\eta/16$, we have

$$
\frac{1}{S}\sum_{s=1}^{S}\frac{1}{K}\sum_{k=0}^{K-1}\left\|\nabla f\left(x_k^s\right)\right\|^2\leq\frac{32R}{SK\eta}+\frac{32}{\eta}J_0+\frac{1}{2}\mu^2L_1^2(d+6)^3.
$$

□

B.4. Complexity analysis

Theorem. *3.2 In Algorithm 1, for $F(x,\xi)\in C^{1,1}$, let the size of sample set $\mathcal{B}$, B =$O(\min\{n,1/\varepsilon^2\})$, the step $\eta=O(\min\{1/(d^{2/3}B^{1/3}),1/(d^{1/3}B^{2/3})\})$, $\mu\leq O(\varepsilon/(L_1d^{1.5}))$, and the number of inner iteration $K\leq O(1/\max\{d\eta^2,d^{0.5}\eta^{1.5}\})$, Gaussian vectors set $D\geq O(\eta d)$. In order to obtain*

$$
\frac{1}{S}\sum\nolimits_{s=0}^{S-1}\frac{1}{K}\sum\nolimits_{k=0}^{K-1}\left\|\nabla f\left(x_k^s\right)\right\|^2\leq\varepsilon^2,
$$

the total number of $T_{\mathcal{SZO}}$ is at most $O(\frac{1}{\varepsilon^2}\max\left\{d^{2/3}B^{1/3},d^{3/5}B^{2/5}\right\})$, with the number of total iterations $T>O(1/(\varepsilon^2\eta))$.

Proof. Based on the results in Theorem 3.1, in order to obtain

$$
\frac{32R}{SK\eta}+\frac{32}{\eta}J_0+\frac{1}{2}\mu^2L_1^2(d+6)^3.\leq\varepsilon^2,
$$

we separately analysis to obtain the complexity:

- The first term: in order to obtain $\frac{1}{2}\mu^2L_1^2(d+6)^3\leq\frac{1}{3}\varepsilon^2$, we have

$$
\mu\leq 2\varepsilon/(L_1(d+6)^{3/2}). \tag{B.20}
$$

- The second term: in order to obtain

$$
\frac{1}{\eta}J_0=\frac{3}{2}\left(\frac{1}{D}+\frac{(B<n)}{B}+3\right)(L_1+2c_0)\,\mu^2L_1^2(d+6)^3\eta+\left(1+\frac{1}{q}c_0\right)\frac{1}{2}\frac{\mathbb{I}(B<n)}{B}H\leq\frac{\varepsilon^2}{3},
$$

  based on μ in (B.20), $\eta\leq\frac{1}{d^{1/2}L_1}<1$ in (B.15), the first sub-term is satisfied; based on $c_0/q<1/4$ in (B.13) and the upper bound of B in (B.19), we require

$$
B\geq\min\left\{n,\frac{1}{\varepsilon^2}\right\}\&\min\left\{n,\eta d\right\}=\min\left\{\max\left\{\eta d,\frac{1}{\varepsilon^2}\right\},n\right\}. \tag{B.21}
$$

We consider the case [3]

$$B \geq \min\left\{n, \frac{1}{\varepsilon^2}\right\}. \tag{B.22}$$

- The third term: in order to obtain $8\frac{R}{SK\eta} \leq \frac{1}{3}\varepsilon^2$, we should require the number of iteration,

$$T = SK \geq \frac{24R}{\varepsilon^2\eta}. \tag{B.23}$$

Furthermore, denote the total number of $\mathcal{SZO}$: $T_{\mathcal{SZO}} = SS_{\mathcal{SZO}}$, where $S = \frac{T}{K}$ is the number of outer iteration, and $S_{\mathcal{SZO}} = DB + K$ is the number of $\mathcal{SZO}$ for each outer iteration. Thus, we have

$$\begin{aligned}
T_{\mathcal{SZO}} = SS_{\mathcal{SZO}} =& \frac{T}{K}\left(DB + K\right) = T\left(\frac{DB}{K} + 1\right) \\
\geq& \frac{R}{\varepsilon^2\eta}\left(\frac{DB}{K} + 1\right) && \text{(B.24)} \\
\geq& \frac{R}{\varepsilon^2}\left(\frac{1}{K\eta}d\eta B + \frac{1}{\eta}\right) && \text{(B.25)} \\
\geq& \frac{R}{\varepsilon^2}\left(\max\left\{d\eta, d^{0.5}\eta^{0.5}\right\}d\eta B + \frac{1}{\eta}\right) && \text{(B.26)} \\
\geq& \frac{R}{\varepsilon^2}\max\left\{d^{2/3}B^{1/3}, d^{1/3}B^{2/3}\right\}, && \text{(B.27)}
\end{aligned}$$

where (B.24) is from the bound of T in (B.23); (B.25) is from the bound of D in (B.19); (B.26) is from the bound of K in (B.16); (B.27) is from following analysis, consider the function $\max\{d\eta, d^{0.5}\eta^{0.5}\}d\eta n + \frac{1}{\eta}$ and $\eta \leq \frac{1}{d^{1/2}}$ in (B.15), we have

- If $\frac{1}{d} \leq \eta \leq \frac{1}{d^{1/2}}$, then

  $$\max\left\{d\eta, d^{1/2}\eta^{1/2}\right\}d\eta B + \frac{1}{\eta} = (d\eta)\,d\eta B + \frac{1}{\eta} \geq d^{2/3}B^{1/3}, \eta^* = \frac{1}{d^{2/3}B^{1/3}},$$

  For the difference of B,

  - $B \leq d \Rightarrow \frac{1}{d} \leq \eta = \frac{1}{d^{2/3}B^{1/3}} \leq \frac{1}{d^{1/2}} \Rightarrow (d\eta)\,d\eta B + \frac{1}{\eta} \geq 2d^{2/3}B^{1/3}$,
  - $B > d \Rightarrow \eta^* = \frac{1}{d^{2/3}B^{1/3}} \leq \frac{1}{d} \leq \frac{1}{d^{1/2}}$, then we set $\eta = \frac{1}{d}$, $\Rightarrow (d\eta)\,d\eta B + \frac{1}{\eta} \geq 2\left(B + d\right)$.

- If $\eta \leq \frac{1}{d}$, because $D \geq \eta d$, thus, we set $D = 1$, then

  $$\max\left\{d\eta, d^{1/2}\eta^{1/2}\right\}B + \frac{1}{\eta} = \left(d^{1/2}\eta^{1/2}\right)B + \frac{1}{\eta} \geq 2d^{1/3}B^{2/3}, \eta^* = \frac{1}{d^{1/3}B^{2/3}}.$$

  For the difference of B,

  - $B \leq d \Rightarrow \eta = \frac{1}{d} \leq \eta^* = \frac{1}{d^{1/3}B^{2/3}}$, $\left(d^{1/2}\eta^{1/2}\right)d\eta B + \frac{1}{\eta} = Bd + d \geq d$;
  - $B > d \Rightarrow \eta = \eta^* = \frac{1}{d^{1/3}B^{2/3}} \leq \frac{1}{d}$, $\left(d^{1/2}\eta^{1/2}\right)d\eta B + \frac{1}{\eta} = 2d^{1/3}B^{2/3} \leq B$.

  Based on the above analysis, we conclude that

  $$\max\left\{d\eta, d^{0.5}\eta^{0.5}\right\}d\eta B + \frac{1}{\eta} = \max\left\{d^{2/3}B^{1/3}, d^{1/3}B^{2/3}\right\},$$

  under the requirement of $\eta \leq \frac{1}{d^{1/2}}$ in (B.15).

[3] otherwise, d is much large than $\frac{1}{\varepsilon^8}$. Because, we consider the case, if $n \geq \eta d \geq \frac{1}{\varepsilon^2}$, then the bast complexity is $T_{SZO} = \frac{d^{3/4}}{\varepsilon^2}$ when $\eta^* = \frac{1}{d^{3/4}}$, which is from (B.26), then $\eta d = d^{1/4} \geq \frac{1}{\varepsilon^2} \Rightarrow d \geq \frac{1}{\varepsilon^8}$, and also require $n \geq \frac{1}{\varepsilon^8}$, if $\varepsilon \leq 0.01$, it will be the extreme case we do not consider here. but even in that case, our result is also better than RGF and RSG.

$\square$

Theorem. *3.3 In Algorithm 2, under Assumption 2.1, for $F(x,\xi) \in C^{1,1}$, let the size of the sample set $\mathcal{B}$, B =$O(\min\left\{n, 1/\varepsilon^2\right\})$, the step η =$O(\min\{b_0^{1/3}/(d^{2/3}B^{1/3}), b_0^{2/3}/(d^{1/3}B^{2/3})\})$, $\mu \leq O(\varepsilon/(L_1 d^{1.5}))$, and the number of inner iteration $K \leq O(1/\max\{d\eta^2, d^{0.5}\eta^{1.5}\})$, Gaussian vectors set $D \geq O(\eta d)$. In order to obtain*

$$\frac{1}{S}\sum\nolimits_{s=0}^{S-1}\frac{1}{K}\sum\nolimits_{k=0}^{K-1}\left\|\nabla f\left(x_k^s\right)\right\|^2 \leq \varepsilon^2,$$

the total number of $T_{\mathcal{SZO}}$ is at most $O(\max\{d^{2/3}B^{1/3}b_0^{2/3}, d^{1/3}B^{2/3}b_0^{1/3}\})$, with number of total iterations $T > O(1/(\varepsilon^2\eta))$.

Proof. Based on the results in Theorem 3.1, the former proof is the same as Theorem 3.2, the difference lies in the optimal value of η. The number of $\mathcal{SZO}$ for each outer iteration becomes $S_{\mathcal{SZO}} = DB + Kb_0$, Thus, we have

$$\begin{aligned}
T_{\mathcal{SZO}} = SS_{\mathcal{SZO}} =& \frac{T}{K}\left(DB + Kb_0\right) = T\left(\frac{DB}{K} + b_0\right) \\
\geq& \frac{R}{\varepsilon^2\eta}\left(\frac{DB}{K} + b_0\right) && \text{(B.28)} \\
\geq& \frac{R}{\varepsilon^2}\left(\frac{1}{K\eta}d\eta B + \frac{b_0}{\eta}\right) && \text{(B.29)} \\
\geq& \frac{R}{\varepsilon^2}\left(\max\left\{d\eta, d^{0.5}\eta^{0.5}\right\}d\eta B + \frac{b_0}{\eta}\right) && \text{(B.30)} \\
\geq& \frac{R}{\varepsilon^2}\max\left\{d^{2/3}B^{1/3}b_0^{2/3}, d^{1/3}B^{2/3}b_0^{1/3}\right\}, && \text{(B.31)}
\end{aligned}$$

where (B.28) is from the bound of T in (B.23); (B.29) is from the bound of D in (B.19); (B.30) is from the bound of K in (B.16); (B.31) is from the following analysis, consider the function $\max\{d\eta, d^{0.5}\eta^{0.5}\}d\eta n + \frac{b_0}{\eta}$ and $\eta \leq \frac{1}{d^{1/2}}$ in (B.15), we have

- If $\frac{1}{d} \leq \eta \leq \frac{1}{d^{1/2}}$, then

$$\max\left\{d\eta, d^{1/2}\eta^{1/2}\right\}d\eta B + \frac{b_0}{\eta} = (d\eta)\,d\eta B + \frac{b_0}{\eta} \geq d^{2/3}B^{1/3}b_0^{2/3}, \eta^* = \frac{b_0^{1/3}}{d^{2/3}B^{1/3}},$$

  consider the size of B:

  - $\frac{B}{b_0} \leq d \Rightarrow \frac{1}{d} \leq \eta = \frac{b_0^{1/3}}{d^{2/3}B^{1/3}} \leq \frac{1}{d^{1/2}} \Rightarrow (d\eta)\,d\eta n + \frac{b_0}{\eta} \geq 2d^{2/3}B^{1/3}b_0^{2/3}$,
  - $\frac{B}{b_0} > d \Rightarrow \eta^* = \frac{b_0^{1/3}}{d^{2/3}B^{1/3}} \leq \frac{1}{d} \leq \frac{1}{d^{1/2}}$, then we set $\eta = \frac{1}{d}, \Rightarrow (d\eta)\,d\eta B + \frac{b_0}{\eta} \geq 2\left(B + db_0\right)$.

- If $\eta \leq \frac{1}{d}$, because $D \geq \eta d$, thus, we set $D = 1$, then

$$\max\left\{d\eta, d^{1/2}\eta^{1/2}\right\}B + \frac{b_0}{\eta} = \left(d^{1/2}\eta^{1/2}\right)B + \frac{b_0}{\eta} \geq 2d^{1/3}B^{2/3}b_0^{1/3}, \eta^* = \frac{b_0^{2/3}}{d^{1/3}B^{2/3}}.$$

  consider the size of B:

  - $\frac{B}{b_0} \leq d \Rightarrow \eta = \frac{1}{d} \leq \eta^* = \frac{b_0^{1/3}}{d^{1/3}B^{2/3}}, \left(d^{1/2}\eta^{1/2}\right)d\eta B + \frac{b_0}{\eta} = B + db_0$;
  - $\frac{B}{b_0} > d \Rightarrow \eta = \eta^* = \frac{b_0^{1/3}}{d^{1/3}B^{2/3}} \leq \frac{1}{d}, \left(d^{1/2}\eta^{1/2}\right)d\eta n + \frac{b_0}{\eta} = 2d^{1/3}B^{2/3}b_0^{2/3}$.

  Based on the above analysis, we conclude that

$$\max\left\{d\eta, d^{0.5}\eta^{0.5}\right\}d\eta B + \frac{b_0}{\eta} = \max\left\{d^{2/3}B^{1/3}b_0^{2/3}, d^{1/3}B^{2/3}b_0^{1/3}\right\},$$

  which under the requirement of $\eta \leq \frac{1}{d^{1/2}}$ in (B.15).

$\square$

C. Convergence proof for Non-Smooth function with Gaussian smooth

C.1. Convergence tool

In this section, we focus on Algorithm 1 that apply to Gaussian-smoothed function for non-smooth function, and mainly give the upper bounds for $\mathbb{E}_u\|G_\mu(\tilde{x}_s, u_\mathcal{D}, \xi) - \nabla f_\mu(\tilde{x}_s)\|^2$, $E_u\|G_\mu(x_k, \xi, u) - G_\mu(\tilde{x}_s, \xi, u)\|^2$ and $\mathbb{E}_{i,j,u}\|\tilde{\nabla}_k\|^2$, which are used for analyzing the convergence sequence. Note, we drop the superscript i and k of ξ and u, respectively, for focusing on a single epoch analysis.

Lemma C.1. *In Algorithm 1, for $F(x,\xi) \in C^{0,0}$ and $G_\mu(\tilde{x}_s, u_\mathcal{D}, \xi_\mathcal{B})$ defined in (3.1), we have*

$$E_u\|G_\mu(\tilde{x}_s, u_\mathcal{D}, \xi) - \nabla f_\mu(\tilde{x}_s)\|^2 \leq \frac{1}{D}L_0^2(d+2)^2,$$

Proof. By the definition of $G_\mu(\tilde{x}_s, u_\mathcal{D}, \xi_\mathcal{B})$ defined in (3.1), we have

$$\begin{aligned}
E_u\|G_\mu(\tilde{x}_s, u_\mathcal{D}, \xi_\mathcal{B}) - \nabla f_\mu(\tilde{x}_s)\|^2 \leq & E_u\|G_\mu(\tilde{x}_s, u_\mathcal{D}, \xi_\mathcal{B})\|^2 \\
\leq & \frac{1}{D}\frac{1}{B}\sum_{i=1}^{B} E_u\|G_\mu(\tilde{x}_s, u, \xi_i)\|^2 \\
\leq & \frac{1}{D}L_0^2(d+2)^2,
\end{aligned}$$

which is based on Lemma A.4 that the vector in $u_\mathcal{D}$ is independently, and Lemma 2.1. ☐

Lemma C.2. *In Algorithm 1, for $F(x,\xi) \in C^{0,0}$, $\mu > 0$, and $G_\mu(x, u, \xi)$ defined in (1.2), we have*

$$E_u\|G_\mu(x_k, \xi, u) - G_\mu(\tilde{x}_s, \xi, u)\|^2 \leq \frac{1}{\mu^2}(d+2)L_0^2\|x_k - \tilde{x}_s\|^2.$$

Proof. By the definition of $G_\mu(x_k, \xi, u)$ in 1.2, we have

$$\begin{aligned}
\|G_\mu(x_k, \xi, u) - G_\mu(\tilde{x}_s, \xi, u)\|^2 = & \left\|\frac{F(x_k + \mu u, \xi) - F(x_k, \xi)}{\mu}u - \frac{F(\tilde{x}_s + \mu u, \xi) - F(\tilde{x}_s, \xi)}{\mu}u\right\|^2 \\
= & \frac{\|u\|^2}{\mu^2}(F(x_k + \mu u, \xi) - F(x_k, \xi) - (F(\tilde{x}_s + \mu u, \xi) - F(\tilde{x}_s, \xi)))^2 \\
\leq & \frac{\|u\|^2}{\mu^2}L_0^2\|x_k - \tilde{x}_s\|^2,
\end{aligned}$$

where the last inequality follows from Lipschitz continue of $F(x,\xi)$. Take expectation with respect to u, we have

$$E_u\|G_\mu(x_k, \xi, u) - G_\mu(\tilde{x}_s, \xi, u)\|^2 \leq \frac{1}{\mu^2}(d+2)L_0^2\|x_k - \tilde{x}_s\|^2,$$

where the second inequality is based on Lemma A.2 for $p = 2$. ☐

Similar to Lemma B.4, based on Lemma C.2 and Lemma C.1, we have

Lemma C.3. *In Algorithm 1, for $F(x,\xi) \in C^{1,1}$, $\mu > 0$, and $\tilde{\nabla}_k$ defined in (3.2), we have*

$$\mathbb{E}_{i,j,u}\|\tilde{\nabla}_k\|^2 \leq \frac{1}{\mu^2}(d+2)L_0^2\|x_k - \tilde{x}_s\|^2 + 3\|\nabla f_\mu(x_k)\|^2 + \frac{3}{D}L_0^2(d+2)^2.$$

C.2. Convergence analysis

In this subsection, mainly based on Lemma B.4, smoothness and update of x in Algorithm 1, we give the new sequence of the proposed algorithm: $\mathbb{E}_{i,j}\left[f_\mu\left(x_{k+1}\right)\right]+c_{k+1}\mathbb{E}_{i,j}\|x_{k+1}-\tilde{x}_s\|^2$. In order to obtain the convergence sequence,we provide the formulation of the sequence c_k, w_k and J_k, which is the key parameter in analyzing the convergence and complexity. In Remark B.1 and B.2, we analyze the the parameter's relationship between K, q and η such that these new formed sequence can be converged.

Lemma C.4. *In Algorithm 1, for $F(x,\xi)\in C^{0,0}$, $\mu>0$, $q>0$, we have*

$$\frac{1}{S}\sum_{s=1}^{S}\frac{1}{K}\sum_{k=0}^{K-1}w_k\|\nabla f_\mu\left(x_k^s\right)\|^2\leq\frac{R}{SK}+J_{k+1},$$

where x^* is the optimal value of function $f_\mu(x)$, $R=\max_x\{f_\mu(x)-f_\mu(x_*):f_\mu(x)\leq f_\mu(x_0)\}$, and

$$c_k=\left(1+\eta q+\frac{1}{\mu^2}\left(d+2\right)L_0^2\eta^2\right)c_{k+1}+\frac{L_1\eta^2}{2}\frac{1}{\mu^2}\left(d+2\right)L_0^2, \tag{C.1}$$

$$w_k=\left(1-\frac{1}{q}c_{k+1}\right)\eta-\left(\frac{3L_1}{2}+c_{k+1}\right)\eta^2, \tag{C.2}$$

$$J_k=\frac{L_1\eta^2}{2}\frac{3}{D}L_0^2(d+2)^2+c_k\eta^2\frac{3}{D}L_0^2(d+2)^2. \tag{C.3}$$

C.2.1 Parameters analysis

Remark C.1. *For μ. The relationship between the non-smooth function and the smoothed function is $|f_\mu\left(x\right)-f\left(x\right)|\leq\mu L_0d^{1/2}$ (Theorem 1 [10], Lemma 2.1). In order to bound the gap in approximation by ε, we follow the setting as in [10], and set*

$$\mu\leq\frac{1}{L_0d^{1/2}}\varepsilon, \tag{C.4}$$

such that $|f_\mu\left(x\right)-f\left(x\right)|\leq\varepsilon$

Remark C.2. *Based on Lemma 2.1, we have $L_1(f_\mu)=d^{1/2}L_0/\mu$, then we consider the sequence $\{J_k\}$, $\{u_k\}$ and $\{c_k\}$, and delete the non-related coefficient that do not affect the convergence, and remain d,η,ε,q, then*

$$\begin{aligned}c_k&=\left(1+\eta q+\frac{d^2}{\varepsilon^2}\eta^2\right)c_{k+1}+\frac{d^3}{\varepsilon^2}\eta^2;\\w_0&=\left(1-\frac{1}{q}c_0\right)\eta-\left(\frac{d}{\varepsilon}+c_0\right)\eta^2;\\J_0&=\frac{1}{D}\frac{d^3}{\varepsilon}\eta^2+\frac{1}{D}d^2c_0\eta^2+\left(1+\frac{1}{q}c_{k+1}\right)\frac{1}{2}\eta\frac{\mathbb{I}(B<n)}{B}H,.\end{aligned}$$

Based on the analysis in Theorem 3.2, Remark B.2 and RemarkB.1, we conclude and require that

$$q=\frac{d^{3/2}}{\varepsilon}\eta^{1/2}, \tag{C.5}$$

$$w_0=\eta,c_0<d, \tag{C.6}$$

$$1>\eta q+\frac{d^2}{\varepsilon^2}\eta^2\Rightarrow\eta<\frac{\varepsilon}{d}, \tag{C.7}$$

$$K\leq1/\max\left\{\frac{d^{3/2}}{\varepsilon}\eta^{2/3},\frac{d^2}{\varepsilon^2}\eta^2\right\}=\frac{\varepsilon^2}{d^2\eta^2}, \tag{C.8}$$

$$D\geq\max\left\{\frac{d^3}{\varepsilon^3}\eta,\frac{1}{\varepsilon^2}d^3\eta\right\}=\frac{d^3}{\varepsilon^3}\eta, \tag{C.9}$$

$$B=\min\left\{n,\frac{1}{\varepsilon^2}\right\}. \tag{C.10}$$

C.3. Complexity analysis

Theorem. *3.4 In Algorithm 1, for $F(x,\xi)\in C^{0,0}$, the step $\eta = O(\varepsilon^{5/3}/(d^{5/3}B^{1/3}))$, $\mu \le O(\varepsilon/(L_0 d^{1/2}))$, and the number of inner iteration $K \le O(\varepsilon^2/(d^2\eta^2))$, Gaussian vectors set $D \ge O(\eta d^3/\varepsilon^3)$. In order to obtain*

$$\frac{1}{S}\sum\nolimits_{s=0}^{S-1}\frac{1}{K}\sum\nolimits_{k=0}^{K-1}\left\|\nabla f_\mu\left(x_k^s\right)\right\|^2 \le \varepsilon^2,$$

the total number of $T_{\mathcal{SZO}}$ is $O(d^{5/3}B^{1/3}/\varepsilon^{11/3})$, number of inner iterations $T > O(1/(\varepsilon^2\eta))$.

Proof. Based on the results in Theorem 3.1, in order to obtain

$$\frac{R}{SK} + J_{k+1} \le \varepsilon^2,$$

we separately analysis to obtain the complexity:

Furthermore, denote the total number of $\mathcal{SZO}$: $T_{\mathcal{SZO}} = SS_{\mathcal{SZO}}$, where $S = \frac{T}{K}$ is the number of outer iteration, and $S_{\mathcal{SZO}} = DB + K$ is the number of $\mathcal{SZO}$ for each outer iteration. Thus, we have

$$\begin{aligned}
T_{\mathcal{SZO}} = SS_{\mathcal{SZO}} =& \frac{T}{K}\left(DB+K\right) = T\left(\frac{DB}{K}+1\right) \\
\ge& \frac{R}{\varepsilon^2\eta}\left(\frac{DB}{K}+1\right) && \text{(C.11)} \\
=& \frac{R}{\varepsilon^2}\left(\frac{DB}{K\eta}+\frac{1}{\eta}\right) \\
\ge& \frac{R}{\varepsilon^2}\left(\frac{d^2\eta}{\varepsilon^2}\frac{d^3}{\varepsilon^3}\eta B+\frac{1}{\eta}\right) \Rightarrow \eta^* = \frac{\varepsilon^{5/3}}{d^{5/3}B^{1/3}} && \text{(C.12)} \\
\ge& \frac{R}{\varepsilon^2}\frac{d^{5/3}B^{1/3}}{\varepsilon^{5/3}} = \frac{d^{5/3}B^{1/3}}{\varepsilon^{11/3}}R, && \text{(C.13)}
\end{aligned}$$

where (C.11) is from the bound of T in (B.23); (C.12) is from the bound of K in (C.8) and D in (C.9); (C.13) is from the optimal value of η^*. □

If for mini-batch SZVR, we can obtain the results from Theorem 3.3 and Theorem 3.4

Theorem C.1. *In Algorithm 1, for $F(x,\xi)\in C^{0,0}$, the step $\eta = O(\varepsilon^{5/3}b_0^{1/3}/(d^{5/3}B^{1/3}))$, $\mu \le O(\varepsilon/(L_0 d^{1/2}))$, and the number of inner iteration $K \le O(\varepsilon^2/(d^2\eta^2))$, Gaussian vectors set $D \ge O(\eta d^3/\varepsilon^3)$. In order to obtain*

$$\frac{1}{S}\sum\nolimits_{s=0}^{S-1}\frac{1}{K}\sum\nolimits_{k=0}^{K-1}\left\|\nabla f_\mu\left(x_k^s\right)\right\|^2 \le \varepsilon^2,$$

the total number of $T_{\mathcal{SZO}}$ is $O(d^{5/3}B^{1/3}/(\varepsilon^{11/3}b_0^{1/3}))$, iteration $T > O(1/(\varepsilon^2\eta))$.