ISSN 0965-5425, Computational Mathematics and Mathematical Physics, 2023, Vol. 63, No. 9, pp. 1600–1653. © Pleiades Publishing, Ltd., 2023.
Russian Text © The Author(s), 2023, published in Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, 2023, Vol. 63, No. 9, pp. 1458–1512.

OPTIMAL CONTROL

Gradient-free Federated Learning Methods with l_1 and l_2-randomization for Non-smooth Convex Stochastic Optimization Problems

B. A. Alashqar[a,*], A. V. Gasnikov[a,b,c], D. M. Dvinskikh[d], and A. V. Lobanov[a,e,f,‡]

[a] *Moscow Institute of Physics and Technology, Dolgoprudny, Russia*
[b] *Institute for Information Transmission Problems RAS, Moscow, Russia*
[c] *Caucasus Mathematical Center, Adyghe State University, Maikop, Russia*
[d] *National Research University Higher School of Economics, Moscow, Russia*
[e] *ISP RAS Research Center for Trusted Artificial Intelligence, Moscow, Russia*
[f] *Moscow Aviation Institute, Moscow, Russia*
**e-mail: lobbsasha@mail.ru*

Received November 18, 2022; revised May 20, 2023; accepted May 29, 2023

Abstract—This paper studies non-smooth problems of convex stochastic optimization. Using the smoothing technique based on the replacement of the function value at the considered point by the averaged function value over a ball (in l_1-norm or l_2-norm) of a small radius centered at this point, and then the original problem is reduced to a smooth problem (whose Lipschitz constant of the gradient is inversely proportional to the radius of the ball). An essential property of the smoothing used is the possibility of calculating an unbiased estimation of the gradient of a smoothed function based only on realizations of the original function. The obtained smooth stochastic optimization problem is proposed to be solved in a distributed federated learning architecture (the problem is solved in parallel: nodes make local steps, e.g. stochastic gradient descent, then communicate—all with all, then all this is repeated). The goal of the article is to build on the basis of modern achievements in the field of gradient–free non-smooth optimization and in the field of federated learning gradient-free methods for solving problems of non-smooth stochastic optimization in the architecture of federated learning.

Keywords: gradient-free methods, inexact oracle, federated learning

DOI: 10.1134/S0965542523090026

1. INTRODUCTION

The article deals with problems of stochastic optimization

$$\min_{x\in Q\subseteq\mathbb{R}^d} f(x) := \mathbb{E}_\xi\left[f(x,\xi)\right], \tag{1}$$

and their saddle generalizations. In this case, it is assumed that the function $f(x)$—convex, non-smooth, and has a bounded Lipschitz constant. It is also assumed that the realization of the function $f(x,\xi)$ is available for observation, but not its gradient (in x) $\nabla f(x,\xi)$. Moreover, the paper also considers the case when this implementation is available with a small noise of a non-random nature.This article is based on the work [1], in which an optimal algorithm is proposed (up to logarithmic factor in the dimension of the space of multipliers in the estimates of oracle calls) for solving problem (1) according to three criteria at once: (1) number of oracle calls (calculations of $f(x,\xi)$), (2) the number of consecutive iterations of the method (you can call the oracle multiple times in one iteration), (3) the maximum allowable level of (potentially adversarial) noise at which it is still possible to achieve the desired accuracy. The approach [1] is based on a rather old idea (see, for example, [2]) of replacing the original non-smooth function $f(x)$ with its smoothed version $f_\gamma(x) = \mathbb{E}_{\tilde{e}}\left[f(x+\gamma\tilde{e})\right]$, where $\tilde{e}$ is a uniformly distributed random vector on $B_2^d(1)$—the Euclidean unit ball centered at zero. For a smoothed function, the final approximation (with

‡ The main contribution to the article belongs to Aleksandr Lobanov ⟨lobbsasha@mail.ru⟩. According to the rules of the journal, the authors of the article are arranged in alphabetical order.

a step γ) of the derivative with respect to a random direction (chosen with equal probability on the Euclideanunit sphere) will be an unbiased (randomized) estimate of the gradient, which, when using a symmetric difference approximation, has a good variance [3]—the same as if $f(x)$ was smooth.

In another recent paper [4] it was shown that if smoothing is used not over the Euclidean ball, but over the ball in the l_1-norm, then in certain situations it is possible to improve the estimates [1] for the number of oracle calls by a logarithmic factor (over the space dimension). However, the question of the number of consecutive iterations was not discussed in [4] and another (narrower) concept of adversarial noise was used, for which the question of exact lower estimates, as far as we know, is still open.

In this paper, we show how the results of [1] can be theoretically improved using the smoothing scheme from [4] (see also the earlier paper [5], where this smoothing scheme was also proposed, but the analysis was performed less accurately). As noted above, the improvements are logarithmic in scale. Note that at the same time the numerical experiments carried out in this work could not clearly capture the effect of such an improvement in estimates.

Another important area of development of the work in [1] was the generalization of all the results (and its modification with smoothing on a ball in the l_1-norm) to distributed algorithms in the architecture of federated learning [6]. Note that, in federated learning, problems of type (1) are considered, firstly, smooth, and secondly, with a full-gradient (stochastic) oracle. Problem (1) is solved independently at each node, then the nodes communicate and calculate the average (over the nodes), then they start solving the problem independently again, starting from this average. After some time, the nodes communicate again, calculate the average, and the process repeats... In this work, by using a smoothing scheme (with the introduction of additional randomization), it is possible to reduce a non-smooth stochastic optimization problem with a gradient-free oracle to a smooth stochastic optimization problem with an oracle, producing a stochastic gradient, in which the stochastics is formed from the initial stochastics inherent in the initial formulation of the problem, and the stochastics that arose during smoothing (randomized). It turned out that in the current literature there are practically no methods of federated learning for non-smooth problems. Apparently, this was due to the fact that, in the general case, it is impossible to perform batch parallelization (parallelization in ξ) for non-smooth problems. However, for a gradient-free oracle, batch parallelization is possible due to the appearance of an additional (randomized) noise in the form of a random direction [1]! Actually, due to this, it is possible to transfer the results of [1] to the architecture of federated learning. To the best of our knowledge, the results obtained in this direction are currently unrivaled, so a literature review of competing works is not presented here.

2. MAIN CONTRIBUTION AND STRUCTURE

The main contribution of the article is as follows.

• We give a detailed description of two smoothing schemes (in parallel): with l_1 and l_2-randomizations. We find the Lipschitz constant of the gradient L_{f_γ} for the l_1-randomization. To explicitly describe the estimation of the second moment in l_2-randomization with two-point feedback, we find the constant c, which was not calculated in the original article [3], from which this constant was taken. We obtain an estimate of the variance (of the second moment) for the one-point case of l_1-randomization. We show that l_1-randomization with a one-point oracle is also superior to l_2-randomization up to $\ln d$, as well as with a two-point oracle.

• Obtaining optimal upper bounds for the rate of convergence of the first-order Minibatch SMP and Single-Machine SMP algorithms for solving saddle-point optimization problems in the federated learning architecture.

• We describe a technique for generating gradient-free algorithms (solutions of saddle-point problems and problems of convex optimization) that are optimal in terms of the number of communication rounds N, the maximum value of admissible noise Δ and oracle complexity T. We show that one-point and two-point algorithms in the federated learning architecture that use l_1-randomization perform better than algorithms using l_2-randomization, up to a logarithmic factor in l_1-norm. We compare one-point algorithms with two-point algorithms and show that for a solution with ε precision (with respect to the function), one-point algorithms require $O(d/\varepsilon^2)$ more calls to the oracle than two-point algorithms. We analyze the operation of the Minibatch Accelerated SGD algorithm using different smoothing schemes on a numerical experiment.

The article is structured as follows: Sections 3 and 4 provide a brief introduction to smoothing techniques and federated learning, respectively. Subsection 4.2 presents the main result of the work. Section 5 gives the main ideas of the proofs of Theorems 1, 2. Numerical experiments are presented in Section 6.

3. SMOOTHING SCHEMES

The smoothing scheme makes it possible to create gradient-free methods for solving non-smooth problems by modifying first-order algorithms of the same name intended for solving smooth problems. The smoothing scheme was first described in the book [2], where the idea of solving problems by first-order methods is presented, using a gradient-free stochastic oracle instead of a stochastic oracle of the first order, which is obtained through the Stokes theorem. Since then, various smoothing techniques have been invented, but the main idea comes from [2]. In this section, we present two smoothing schemes in parallel: with l_1-randomization [4, 5] and with l_2-randomization [1], which includes stochastic optimization and a biased estimation of a gradient-free oracle. For this, we consider a stochastic non-smooth convex optimization problem

$$\min_{x\in Q\subseteq\mathbb{R}^d}\{f(x) := \mathbb{E}_\xi[f(x,\xi)]\}, \tag{2}$$

where Q is a convex and compact set and $f(x,\xi)$ is a convex function on the set $Q_\gamma := Q + B_p^d(\gamma)$. Here it is assumed that the gradient-free oracle returns the value of the function $f(x)$, possibly, with some adversarial noise $\delta(x)$: $f_\delta(x) := f(x) + \delta(x)$.

3.1. Notation and Assumptions

We denote by $\langle x, y\rangle := \sum_{i=1}^d x_i y_i$ the standard scalar product of $x, y \in \mathbb{R}^d$. By $\|x\|_p := \left(\sum_{i=1}^d |x_i|^p\right)^{1/p}$ we denote l_p-norm ($p \geq 1$) in $\mathbb{R}^d$ and $B_p^d(r) := \left\{x \in \mathbb{R}^d : \|x\|_p \leq r\right\}$, $S_p^d(r) := \left\{x \in \mathbb{R}^d : \|x\|_p = r\right\}$ define l_p-ball and l_p-sphere respectively. The volume of the l_p-ball is determined by

$$V(B_p^d(r)) := c_d r^d = \frac{\left(2\Gamma\left(\frac{1}{p}+1\right)\right)^d}{\Gamma\left(\frac{d}{p}+1\right)} r^d,$$

where $\Gamma(\cdot)$ stands for the gamma-function. To denote the "distance" between the initial point x^0 and the solution of the original problem x_* we introduce $R := \tilde{O}\left(\left\|x^0 - x_*\right\|_p\right)$, where $\tilde{O}(\cdot)$ is $O(\cdot)$ up to the logarithmic factor in $\sqrt{\ln d}$.

Assumption 1 (Lipschitz continuous function). The function $f(x,\xi)$ is an *M-Lipschitz* continuous function in the l_p-norm, that is, for all $x, y \in Q$ we have

$$|f(y,\xi) - f(x,\xi)| \leq M(\xi)\|y - x\|_p.$$

Moreover, there is a positive constant M, which is defined as follows: $\mathbb{E}[M^2(\xi)] \leq M^2$. In particular, for $p = 2$ we use the notation M_2 for the Lipschitz constant.

Assumption 2 (Bounded noise). For all $x \in Q$ we have $|\delta(x)| \leq \Delta$, where Δ is the level of noise.

Assumption 3. For all $x \in Q$ we have $\mathbb{E}_\xi[|f(x,\xi)|^2] \leq G^2$.

3.2. Smooth Approximation

Since problem (2) is non-smooth, we introduce the following smooth approximation of a non-smooth function

$$f_\gamma(x) := \mathbb{E}_{\tilde{e}}[f(x + \gamma\tilde{e})], \tag{3}$$

where $\gamma > 0$, $\tilde{e}$ is a random vector uniformly distributed on $B_p^d(1)$ (below, we restrict ourselves to the cases $p = 1$ and $p = 2$). Here $f(x) := \mathbb{E}f(x, \xi)$.

The following lemma presents a set of properties of approximation of the function f depending on the distribution of the vector $\tilde{e}$. Based on [1, 4, 7] and substituting the found Lipschitz constant of the gradient L_{f_γ} in the case $p = 1$, we write down the properties of the function f_γ.

Lemma 1. *For all $x, y \in Q$ with Assumption* 1*,*

$\tilde{e} \in RB_1^d(1)$	$\tilde{e} \in RB_2^d(1)$
$f(x) \le f_\gamma(x) \le f(x) + \frac{2}{\sqrt{d}}\gamma M_2$	$f(x) \le f_\gamma(x) \le f(x) + \gamma M_2$
f_γ – M-Lipschitz:	
$\lvert f_\gamma(y) - f_\gamma(x)\rvert \le M\lVert y - x\rVert_p$	$\lvert f_\gamma(y) - f_\gamma(x)\rvert \le M\lVert y - x\rVert_p$
f_γ has an L_{f_γ}-Lipschitz gradient:	
$\lVert\nabla f_\gamma(y) - \nabla f_\gamma(x)\rVert_q \le \underbrace{\frac{dM}{2\gamma}}_{L_{f_\gamma}}\lVert y - x\rVert_p$	$\lVert\nabla f_\gamma(y) - \nabla f_\gamma(x)\rVert_q \le \underbrace{\frac{\sqrt{d}M}{\gamma}}_{L_{f_\gamma}}\lVert y - x\rVert_p$

where q is such that $1/p + 1/q = 1$.

The proof is given in Subsection 5.1.

3.3. Randomization with Two-Point Feedback

An approximation of the gradient of the noisy function $f_\gamma(x, \xi)$ from (3) can be obtained through two points close to x. To do this, we define a random vector e, uniformly distributed on $S_p^d(1)$. Then the gradient can be estimated by the following approximation.

- Gradient estimate for l_1-randomization ($e \in RS_1^d(1)$) [5] (see also [4]):

$$\nabla f_\gamma(x, \xi, e) = \frac{d}{2\gamma}(f_\delta(x + \gamma e, \xi) - f_\delta(x - \gamma e, \xi))\operatorname{sgn}(e). \tag{4}$$

- Gradient estimate for l_2-randomization ($e \in RS_2^d(1)$) [3]:

$$\nabla f_\gamma(x, \xi, e) = \frac{d}{2\gamma}(f_\delta(x + \gamma e, \xi) - f_\delta(x - \gamma e, \xi))e. \tag{5}$$

To evaluate the gradient in (4) and (5), a central finite-difference randomization scheme was chosen, since in [8] it is explained that in the smooth case, it is more advantageous to evaluate the gradient using the central finite-difference scheme (CFD), rather than the forward finite difference scheme (FFD). Not that for $\Delta = 0$ the estimates will be unbiased, i.e. $E_{e,\xi}\left[\nabla f_\gamma(x, \xi, e)\right] = \nabla f_\gamma(x)$.

Next, we present the properties of $\nabla f_\gamma(x, \xi, e)$ for two randomizations using well-known results from [1, 3–5, 9, 10]. In many papers, for l_2-randomization, the estimate of the second moment is written up to the constant c, therefore, in Lemma 2, estimates of the second moment for l_1 and l_2-randomizations are given with a refinement of the constant c.

Lemma 2. *For all $x \in Q$ with Assumptions* 1 *and* 2 *we have*

(i) $\nabla f_\gamma(x, \xi, e)$ *with l_1-randomization* (4) *has the variance estimate* (*second moment*)*:*

$$E_e\left[\lVert\nabla f_\gamma(x, \xi, e)\rVert_q^2\right] \le \kappa(p, d)\left(M_2^2 + \frac{d^2\Delta^2}{12(1 + \sqrt{2})^2\gamma^2}\right),$$

where $1/p + 1/q = 1$ *and*

$$\kappa(p, d) = 48(1 + \sqrt{2})^2 d^{2 - \frac{2}{p}}.$$

If Δ *is small enough, then*

$$E_e\left[\left\|\nabla f_\gamma(x,\xi,e)\right\|_q^2\right] \le 2\kappa(p,d)M_2^2. \tag{6}$$

(ii) $\nabla f_\gamma(x,\xi,e)$ *with* l_2*-randomization* (5) *has the variance estimate* (*second moment*):

$$E_e\left[\left\|\nabla f_\gamma(x,\xi,e)\right\|_q^2\right] \le \kappa(p,d)\left(M_2^2 + \frac{d^2\Delta^2}{\sqrt{2}\gamma^2}\right),$$

where $1/p + 1/q = 1$ *and*

$$\kappa(p,d) = \sqrt{2}\min\{q, \ln d\} d^{2-\frac{2}{p}}.$$

If Δ *is small enough, then*

$$E_e\left[\left\|\nabla f_\gamma(x,\xi,e)\right\|_q^2\right] \le 2\kappa(p,d)M_2^2. \tag{7}$$

The proof is given in Subsection 5.2.

3.4. Randomization with Single-Point Feedback

For the case where two-point feedback is not available, smoothing schemes can use one-point feedback through the following unbiased estimate:

- Gradient estimate for l_1-randomization ($e \in RS_1^d(1)$):

$$\nabla f_\gamma(x,\xi,e) = \frac{d}{\gamma}(f_\delta(x+\gamma e,\xi))\operatorname{sgn}(e). \tag{8}$$

- Gradient estimate for l_2-randomization ($e \in RS_2^d(1)$) [1]:

$$\nabla f_\gamma(x,\xi,e) = \frac{d}{\gamma}(f_\delta(x+\gamma e,\xi))e. \tag{9}$$

Then the properties of $\nabla f_\gamma(x,\xi,e)$ for the two randomizations will have the following form.

Lemma 3. *For all* $x \in Q$ *with Assumptions 2 and 3 we have*

(i) $\nabla f_\gamma(x,\xi,e)$ *with* l_1*-randomization* (8) *has the variance estimate* (*second moment*):

$$E_e\left[\left\|\nabla f_\gamma(x,\xi,e)\right\|_q^2\right] \le \kappa(p,d)\left(\frac{G^2}{\gamma^2} + \frac{\Delta^2}{\gamma^2}\right),$$

where $1/p + 1/q = 1$ *and*

$$\kappa(p,d) = d^{4-\frac{2}{p}}.$$

If Δ *is small enough, then*

$$E_e\left[\left\|\nabla f_\gamma(x,\xi,e)\right\|_q^2\right] \le 2\kappa(p,d)\frac{G^2}{\gamma^2}. \tag{10}$$

(ii) $\nabla f_\gamma(x,\xi,e)$ *with* l_2*-randomization* (9) *has the variance estimate* (*second moment*):

$$E_e\left[\left\|\nabla f_\gamma(x,\xi,e)\right\|_q^2\right] \le \kappa(p,d)\left(\frac{G^2}{\gamma^2} + \frac{\Delta^2}{\gamma^2}\right),$$

where $1/p + 1/q = 1$ *and*

$$\kappa(p,d) = \min\{q, \ln d\} d^{3-\frac{2}{p}}.$$

If Δ is small enough, then

$$E_e\left[\left\|\nabla f_\gamma(x,\xi,e)\right\|_q^2\right] \le 2\kappa(p,d)\frac{G^2}{\gamma^2}. \tag{11}$$

The proof is given in Subsection 5.3.

3.5. Smoothing Algorithm

Based on the above elements, we will describe a general approach, referred to as the Smoothing Scheme. Suppose we have some accelerated batch algorithm $\mathbf{A}(L,\sigma^2)$ of federated learning architecture that solves problem (2) with the assumption that f is smooth and satisfies

$$\left\|\nabla f(y) - \nabla f(x)\right\|_q \le L\left\|y - x\right\|_p \quad \forall x, y \in Q_\gamma,$$

and using a first-order stochastic oracle that depends on the random variable η and returns at the point x the biased stochastic gradient $\nabla_x f_\gamma(x,\eta)$

$$\mathbb{E}_\eta\left[\left\|\nabla_x f_\gamma(x,\eta) - \nabla f(x)\right\|_q^2\right] \le \sigma^2.$$

Then the general approach of the smoothing scheme is to apply $\mathbf{A}(L,\sigma^2)$ to the smoothed problem

$$\min_{x\in Q\subseteq\mathbb{R}^d} f_\gamma(x), \tag{12}$$

for a solution with $\varepsilon/2$ accuracy with known parameters $\eta = e$, $\nabla_x f_\gamma(x,\eta) = \nabla f_\gamma(x,\xi,e)$, $L = L_{f_\gamma}$, and γ, presented in Corollary 1.

Corollary 1. According to Lemma 1, in order to obtain the ε-accuracy of the solution of problem (2), it is necessary to solve problem (12) with $\varepsilon/2$-accuracy with the following parameter

Smoothing scheme with l_1-randomization	Smoothing scheme with l_2-randomization
$\gamma = \dfrac{\sqrt{d}\varepsilon}{4M_2}$	$\gamma = \dfrac{\varepsilon}{2M_2}$

where $\varepsilon > 0$ is the desired accuracy of the solution to problem (2) in terms of the suboptimality: $\mathbb{E}[f(x^N) - f(x_*)] \le \varepsilon$.

Corollary 2. According to Lemma 1, substituting the parameter γ from Corollary 1, we have

Smoothing scheme with l_1-randomization	Smoothing scheme with l_2-randomization
$L_{f_\gamma} = \dfrac{2\sqrt{d}MM_2}{\varepsilon}$	$L_{f_\gamma} = \dfrac{2\sqrt{d}MM_2}{\varepsilon}$

Corollary 3. According to Lemma 2, equations (6) and (7) for a two-point oracle will take the form

Smoothing scheme with l_1-randomization	Smoothing scheme with l_2-randomization
$\sigma^2 \le 48(1+\sqrt{2})^2 d^{2-\frac{2}{p}} M_2^2$	$\sigma^2 \le 2\sqrt{2}\min\{q, \ln d\} d^{2-\frac{2}{p}} M_2^2$

if Δ is small enough.

Corollary 4. According to Lemma 3, equations (10) and (11) for a single-point oracle will take the form

Smoothing scheme with l_1-randomization	Smoothing scheme with l_2-randomization
$\sigma^2 \le 32 d^{3-\frac{2}{p}} \dfrac{G^2 M_2^2}{\varepsilon^2}$	$\sigma^2 \le 8\min\{q, \ln d\} d^{3-\frac{2}{p}} \dfrac{G^2 M_2^2}{\varepsilon^2}$

if Δ is small enough.

Remark 1. If, instead of a stochastic non-smooth convex optimization problem (2), we consider a stochastic non-smooth convex-concave saddle-point problem

$$\min_{x\in Q_x\subseteq\mathbb{R}^{d_x}}\max_{y\in Q_y\subseteq\mathbb{R}^{d_y}}\{f(x,y):=\mathbb{E}[f(x,y,\xi)]\},$$

then applying the smoothing scheme described above in this section separately to the x, and y variables, we obtain the same results as for convex optimization with the corresponding changes in $f(x,\xi)$ by $f(z,\xi)$, where $z:=(x,y), z\in Q_z:=Q_x\times Q_y$, except for one point in Lemma 1:

- $(\tilde{e}\in RB_1^d(1))$ instead of

$$f(x)\le f_\gamma(x)\le f(x)+\frac{2}{\sqrt{d}}\gamma M_2.$$

We have

$$f(x,y)-\frac{2}{\sqrt{d}}\gamma_y M_{2,y}\le f_\gamma(x,y)\le f(x,y)+\frac{2}{\sqrt{d}}\gamma_x M_{2,x};$$

- $(\tilde{e}\in RB_2^d(1))$ instead of

$$f(x)\le f_\gamma(x)\le f(x)+\gamma M_2.$$

We have

$$f(x,y)-\gamma_y M_{2,y}\le f_\gamma(x,y)\le f(x,y)+\gamma_x M_{2,x},$$

where $\gamma=(\gamma_x,\gamma_y)$, $M_{2,x}$ and $M_{2,y}$ are the corresponding Lipschitz constants in the l_2-norm.

4. FEDERATED LEARNING

The architecture of distributed learning looks like this: a data set is distributed between "computers", each computer makes one local update (a local step, for example, a step of stochastic gradient descent), after which a global update occurs (a global step, communication of all computers with all), then the chain of local-global updates repeats. However, a global update, for example, when the data size is large, consumes a lot of computing resources, unlike a local update. Then the federated learning architecture is introduced, shown in Fig. 1, where in the homogeneous case B computers do in parallel K local updates before each communication round, the total number of which is N. Thus, NK is the total number of iterations of the algorithm, and $T=NKB$ is the total number of stochastic gradient calls.

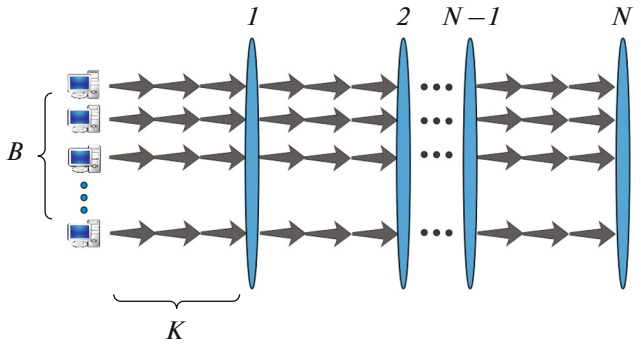

Fig. 1. Architecture of federated learning.

4.1. Optimal First-Order Algorithms

In this section, we will stop and consider a class of first-order accelerated methods, namely Minibatch Accelerated SGD (Mb-Ac-SGD) and Single-Machine Accelerated SGD (SM-Ac-SGD) from [11], Local-AC-CA from [12] and FedAc from [13], whose convergence rate results are presented in Table 1.

Table 1. Convergence rate results

Algorithm	$\mathbb{E}[f(\cdot)] - f^* \lesssim \ldots$	Reference
Mb-Ac-SGD	$\frac{LR^2}{N^2} + \frac{\sigma R}{\sqrt{BNK}}$	(Woodworth et al., 2021) [11]
SM-Ac-SGD	$\frac{LR^2}{N^2K^2} + \frac{\sigma R}{\sqrt{NK}}$	(Woodworth et al., 2021) [11]
Local-AC-CA	$\frac{LR^2}{N^2K^2} + \frac{\sigma R}{\sqrt{BNK}}$	(Woodworth et al., 2020) [12]
FedAc	$\frac{LR^2}{N^2K} + \frac{\sigma R}{\sqrt{BNK}} + \min\left\{\frac{L^{1/3}\sigma^{2/3}R^{4/3}}{NK^{1/3}}, \frac{L^{1/2}\sigma^{1/2}R^{3/2}}{NK^{1/4}}\right\}$	(Yuan, Ma, 2020) [13]
Mb-SMP	$\max\left\{\frac{LR^2}{N}, \frac{\sigma R}{\sqrt{BNK}}\right\}$	Appendix A
SM-SMP	$\max\left\{\frac{LR^2}{NK}, \frac{\sigma R}{\sqrt{NK}}\right\}$	Appendix A

The following notation was used in the table: R is $\left\|x^0 - x_*\right\|_2$; B is the number of working computers; K is the number of local updates; N is the number of communication rounds; L is smoothness.

For a quadratic objective function, it was proved that $K > 1$ local update leads to optimal convergence rate estimates (the Local-AC-CA algorithm from [12]). The FedAc algorithm from [13] generalizes the results of [12] to the case of convex functions.

But already in 2021, optimal convergence rate estimates were obtained in [11]. This article states that optimal estimates can be obtained only in two cases. The first case (the Minibatch Accelerated SGD algorithm) assumes that each computer performs one local update before a communication round. The second case (Single-Machine Accelerated SGD algorithm) assumes that only one computer is running, which performs NK updates. Despite the proved results of [11], in practice, in the general case of convex functions, it turns out to use $K > 1$ local steps, while losing slightly in accuracy, but significantly gaining in computational resources. It is the practical results that allow us to expect positive theoretical results in the future.

Existing first-order algorithms for federated learning architecture often solve the convex optimization problem. As for saddle-point optimization problems, there are currently no algorithms in the federated learning architecture. In this paper, we have developed and obtained optimal convergence rate estimates for the first-order methods, namely, Minibatch SMP (Mb-SMP) and Single-Machine SMP (SM-SMP) for solving saddle-point optimization problems using a similar approach as in convex optimization [11]. Convergence estimates of algorithms for solving saddle-point problems are also given in Table 1. A detailed description of obtaining upper convergence rate estimates for the Minibatch SMP (Mb-SMP) and Single-Machine SMP (SM-SMP) algorithms is given in Appendix A.

4.2. Optimal Zero-Order Algorithms

This subsection presents the main result of this article, which is to combine the two global optimization ideas presented in Section 3 and Subsection 4.1. Namely, the solution of stochastic non-smooth convex/convex-concave optimization problems by gradient-free algorithms of the federated-learning architecture. To develop and apply gradient-free methods for solving non-smooth problems, it is proposed to choose a first-order method used to solve smooth problems in the architecture of federated learning. Next, it is necessary to modify the algorithm of the chosen method by replacing the calculation of the stochastic gradient with a gradient-free approximation with l_1 (4) or l_2 (5) randomization. The resulting gradient-free algorithm will have the same name as the first order algorithm, but it will not require information about the stochastic gradient.

Let the algorithm $\mathbf{A}(L, \sigma^2)$ be understood as an accelerated batch algorithm for solving problems of convex optimization: Minibatch and Single-Machine Accelerated SGD, Local AC-CA and FedAc and

accelerated batch algorithms for solving problems with a saddle-point: Minibatch SMP and Single-Machine SMP in federated-learning architecture. Then, in Theorems 1 and 2, we assume that this property holds: the algorithm $\mathbf{A}(L,\sigma^2)$ (with a biased gradient-free oracle $\nabla f_\gamma(x,\xi,e)$) is reliable if the bias from Corollary 5 for l_1-randomization, and from Corollary 6 for l_2-randomization (discussed in Section 5.4) does not accumulate during the iteration of the method. That is, if for $\mathbf{A}(L,\sigma^2)$ with $\Delta = 0$:

$$E[f_\gamma(x^N) - f(x_*)] \le \Theta_A(N),$$

then

- for $\Delta > 0$ and $d^{4-\frac{2}{p}}\Delta^2\gamma^{-2} \lesssim \kappa(p,d)M_2^2$ from (6):

$$E[f_\gamma(x^N) - f(x_*)] \le O\left(\Theta_A(N) + d\Delta R\gamma^{-1}\right); \tag{13}$$

- for $\Delta > 0$ and $d^2\Delta^2\gamma^{-2} \lesssim dM_2^2$ from (7):

$$E[f_\gamma(x^N) - f(x_*)] \le O\left(\Theta_A(N) + \sqrt{d}\Delta R\gamma^{-1}\right). \tag{14}$$

The assumption about the fulfillment of this property is based on the article [14], where the convergence analysis of methods for biased stochastic oracles was developed. Therefore, in Theorems 1, 2 we present the results, assuming that it is possible by analogy to analyze the convergence of the methods of federated learning considered in this paper for biased stochastic oracles. However, in the proof of Theorems 1 and 2, we will consider the case with $\Delta = 0$ and give optimal estimates for the parameters of the developed gradient-free algorithms.

Theorem 1. *The smoothing scheme from Section* 3 *applied to problem* (2) *ensures the convergence of the following two-point gradient-free algorithms: Minibatch and Single Machine Accelerated SGD* [11]*, Local-AC-SA* [12]*, and FedAc* [13]*. In other words, to achieve* ε *accuracy of solving problem* (2)*, it is necessary to perform* NK *iterations with the maximum allowable level of noise* Δ *and the total number of calls to the gradient-free oracle* T *in accordance with the chosen method and smoothing scheme:*

- *Minibatch Accelerated SGD*

(i) *for* l_1*-randomization* (4):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = O\left(\frac{d^{1/4}\sqrt{MM_2}R}{\varepsilon}\right); \quad K = 1; \quad B = O\left(\frac{\kappa(p,d)M_2^2R^2}{KN\varepsilon^2}\right);$$

$$T = O\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} O\left(\dfrac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2, \\ O\left(\dfrac{M_2^2R^2}{\varepsilon^2}\right), & p = 1; \end{cases}$$

(ii) *for* l_2*-randomization* (5):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = O\left(\frac{d^{1/4}\sqrt{MM_2}R}{\varepsilon}\right); \quad K = 1; \quad B = O\left(\frac{\kappa(p,d)dM_2^2R^2}{KN\varepsilon^2}\right);$$

$$T = \tilde{O}\left(\frac{\kappa(p,d)dM_2^2R^2}{\varepsilon^2}\right) = \begin{cases} \tilde{O}\left(\dfrac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2, \\ \tilde{O}\left(\dfrac{(\ln d)M_2^2R^2}{\varepsilon^2}\right), & p = 1; \end{cases}$$

• *Single-Machine Accelerated SGD*

(i) *for l_1-randomization* (4):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = 1; \quad K = O\left(\frac{\varkappa(p,d)M_2^2R^2}{\varepsilon^2}\right); \quad B = 1;$$

$$T = O\left(\frac{\varkappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} O\left(\dfrac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2, \\ O\left(\dfrac{M_2^2R^2}{\varepsilon^2}\right), & p = 1; \end{cases}$$

(ii) *for l_2-randomization* (5):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = 1; \quad K = O\left(\frac{\varkappa(p,d)M_2^2R^2}{\varepsilon^2}\right); \quad B = 1;$$

$$T = \tilde{O}\left(\frac{\varkappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} \tilde{O}\left(\dfrac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2, \\ \tilde{O}\left(\dfrac{(\ln d)M_2^2R^2}{\varepsilon^2}\right), & p = 1; \end{cases}$$

• *Local-AC-SA*

(i) *for l_1-randomization* (4):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = 1; \quad K = O\left(\frac{d^{1/4}\sqrt{MM_2}R}{\varepsilon}\right); \quad B = O\left(\frac{\varkappa(p,d)M_2^2R^2}{KN\varepsilon^2}\right);$$

$$T = O\left(\frac{\varkappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} O\left(\dfrac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2, \\ O\left(\dfrac{M_2^2R^2}{\varepsilon^2}\right), & p = 1; \end{cases}$$

(ii) *for l_2-randomization* (5):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = 1; \quad K = O\left(\frac{d^{1/4}\sqrt{MM_2}R}{\varepsilon}\right); \quad B = O\left(\frac{\varkappa(p,d)M_2^2R^2}{KN\varepsilon^2}\right);$$

$$T = \tilde{O}\left(\frac{\varkappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} \tilde{O}\left(\dfrac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2, \\ \tilde{O}\left(\dfrac{(\ln d)M_2^2R^2}{\varepsilon^2}\right), & p = 1; \end{cases}$$

• *Federated Accelerated SGD* (*FedAc*)

(i) *for* l_1*-randomization* (4):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = O\left(\frac{d^{1/6}(\kappa(p,d)M)^{1/3}M_2R^{4/3}}{K^{1/3}\varepsilon^{4/3}}\right); \quad K = O\left(\frac{\kappa(p,d)M_2^2R^2}{BN\varepsilon^2}\right); \quad B = 1;$$

$$T = O\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} O\left(\frac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2, \\ O\left(\frac{M_2^2R^2}{\varepsilon^2}\right), & p = 1; \end{cases}$$

(ii) *for* l_2*-randomization* (5):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = O\left(\frac{d^{1/2}(\kappa(p,d)M)^{1/3}M_2R^{4/3}}{K^{1/3}\varepsilon^{4/3}}\right); \quad K = O\left(\frac{\kappa(p,d)M_2^2R^2}{BN\varepsilon^2}\right); \quad B = 1;$$

$$T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} \tilde{O}\left(\frac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2, \\ \tilde{O}\left(\frac{(\ln d)M_2^2R^2}{\varepsilon^2}\right), & p = 1. \end{cases}$$

A detailed proof of Theorem 1 is given in Appendix B.

According to the results of Theorem 1, it is worth noting that the number of local updates $K > 1$ using $B > 1$ computers with optimal values of N and T was obtained only for one algorithm, namely Local-AC-SA, which is used for a particular quadratic case. This confirms that currently there are only two optimal (in Δ, N and T) algorithms in theory: Minibatch and Single-Machine Accelerated SGD. It was shown in [13] that in practice it is possible to obtain a result in which the algorithm performs in parallel $K > 1$ local updates on every $B > 1$ computers.

In Theorem 2, we also obtain optimal estimates for the parameters considered in Subsection 4.1 of algorithms for solving stochastic non-smooth problems with a saddle-point that are optimal in Δ, N, and T.

Theorem 2. *The smoothing scheme from Section* 3 *applied to the saddle-point problem* (*see Remark* 1) *ensures the convergence of the following two-point gradient-free algorithms: Minibatch SMP and SingleMachine SMP from Appendix* A. *In other words, to achieve* ε *the accuracy of solving the saddle-point problem* (*see Remark* 1)*, it is necessary to perform* NK *iterations with the maximum allow ablelevel of noise* Δ *and the total number of calls to the gradient-free oracle* T *in accordance with the chosen method and smoothing scheme:*

• *Minibatch SMP*

(i) *for* l_1*-randomization* (4):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = 1; \quad K = 1; \quad B = O\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right);$$

$$T = O\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} O\left(\frac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2, \\ O\left(\frac{M_2^2R^2}{\varepsilon^2}\right), & p = 1; \end{cases}$$

(ii) *for l_2-randomization* (5):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = 1; \quad K = 1; \quad B = O\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right);$$

$$T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} \tilde{O}\left(\dfrac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2, \\ \tilde{O}\left(\dfrac{(\ln d)M_2^2R^2}{\varepsilon^2}\right), & p = 1; \end{cases}$$

• *Single-Machine SMP*

(i) *for l_1-randomization* (4):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = 1; \quad K = O\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right); \quad B = 1;$$

$$T = O\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} O\left(\dfrac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2, \\ O\left(\dfrac{M_2^2R^2}{\varepsilon^2}\right), & p = 1; \end{cases}$$

(ii) *for l_2-randomization* (5):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = 1; \quad K = O\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right); \quad B = 1;$$

$$T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} \tilde{O}\left(\dfrac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2, \\ \tilde{O}\left(\dfrac{(\ln d)M_2^2R^2}{\varepsilon^2}\right), & p = 1. \end{cases}$$

A detailed proof of Theorem 2 is given in Appendix C.

According to the results of Theorem 1 and Theorem 2, it is easy to see that for all algorithms the optimal number of calls to a gradient-free oracle in the l_1-norm is $\tilde{O}\left(\dfrac{(\ln d)M_2^2R^2}{\varepsilon^2}\right)$ with l_2-randomization, while for l_1-randomization it equals $O\left(\dfrac{M_2^2R^2}{\varepsilon^2}\right)$, where ε is the accuracy of solving a non-smooth problem. This result confirms that the l_1-randomization smoothing scheme performs better in the federated learning architecture than the l_2-randomization. In Section 6, we will test this result on a numerical experiment.

Remark 2. To obtain the results of Theorems 1 and 2, we used the assumption that a two-point feedback is available (see Subsection 3.3). If one-point feedback is used instead of two-point feedback (see Subsection 3.4), then this will lead to the same iterative complexity (number of communication rounds) N and the maximum level of noise Δ, but the oracle complexity will increase by a factor of $O(d/\varepsilon^2)$. This case is discussed in detail in Appendix D.

Remark 3. It is worth noting that the idea of combining two techniques, namely, federated learning with smoothing schemes, is not limited to the considered first-order algorithms and can be used to solve non-smooth problems by other algorithms used in the federated learning architecture.

5. PROOF SCHEMES

In this section, we present schemes for the proofs of Theorem 1 and Theorem 2. Detailed proofs of these theorems can be found in Appendix B and C. Here we focus on the proof of Lemmas 1–3 and on finding an estimate for the level of noise (adversarial noise).

5.1. Proof of Lemma 1

In this subsection, we consider the non-Euclidean case when the random vector $\tilde{e}$ is uniformly distributed on the l_1-ball. The proof of the Euclidean case can be found in Theorem 2.1 from [1].

For all $x, y \in Q$

(1) $f(x) \le f_\gamma(x) \le f(x) + \frac{2}{\sqrt{d}}\gamma M_2$;

(2) f_γ is M-Lipschitz:

$$\left|f_\gamma(y) - f_\gamma(x)\right| \le M\left\|y - x\right\|_p,$$

(3) f_γ has $L_{f_\gamma} = \frac{dM}{2\gamma}$-Lipschitz gradient:

$$\left\|\nabla f_\gamma(y) - \nabla f_\gamma(x)\right\|_q \le L_{f_\gamma}\left\|y - x\right\|_2 \le L_{f_\gamma}\left\|y - x\right\|_p,$$

where q such that $1/p + 1/q = 1$.

Proof. For the first inequality of the first property, we use the convexity of the function $f(x)$

$$f_\gamma(x) = \mathbb{E}_{\tilde{e}}\left[f(x + \gamma\tilde{e})\right] \ge \mathbb{E}_{\tilde{e}}\left[f(x) + \langle\nabla f(x), \gamma\tilde{e}\rangle\right] = \mathbb{E}_{\tilde{e}}\left[f(x)\right] = f(x).$$

For the second inequality of the first property, applying Lemma 1 from [4], we have:

$$\left|f_\gamma(x) - f(x)\right| = \left|\mathbb{E}_{\tilde{e}}\left[f(x + \gamma\tilde{e})\right] - f(x)\right| \le \mathbb{E}_{\tilde{e}}\left[\left|f(x + \gamma\tilde{e}) - f(x)\right|\right] \le \gamma M_2\mathbb{E}_{\tilde{e}}\left[\left\|\tilde{e}\right\|_2\right] \le \frac{2}{\sqrt{d}}\gamma M_2,$$

using the fact that f is M_2-Lipschitz.

For the second property:

$$\left|f_\gamma(y) - f_\gamma(x)\right| = \left|\mathbb{E}_{\tilde{e}}\left[f(y + \gamma\tilde{e}) - f(x + \gamma\tilde{e})\right]\right| \le \mathbb{E}_{\tilde{e}}\left[\left|f(y + \gamma\tilde{e}) - f(x + \gamma\tilde{e})\right|\right] \le M\left\|y - x\right\|_p.$$

And for the third property, applying Lemma 11 from [15] we have for any $x, y \in Q$,

$$\left\|\nabla f_\gamma(y) - \nabla f_\gamma(x)\right\|_q = \left\|\int\limits_{B_1^d(\gamma)} \nabla f(y + \tilde{e})\mu(\tilde{e})d\tilde{e} - \int\limits_{B_1^d(\gamma)} \nabla f(x + \tilde{e})\mu(\tilde{e})d\tilde{e}\right\|_q$$

$$\le \left\|\int\limits_{Q_\gamma} \nabla f(z)\mu(z - y)dz - \int\limits_{Q_\gamma} \nabla f(z)\mu(z - x)dz\right\|_q \le M\underbrace{\int\limits_{Q_\gamma}\left|\mu(z - y) - \mu(z - x)\right|dz}_{I_1},$$

where the second inequality follows from $z = x + \tilde{e}$ and

$$\mu(x) = \begin{cases}\dfrac{1}{V(B_1^d(\gamma))}, & x \in B_1^d(\gamma),\\ 0, & \text{otherwise.}\end{cases}$$

Next, to evaluate the integral I_1 we apply the same approach as in the proof of Lemma 8 from [16] and consider the cases where $\left\|y - x\right\|_1 > 2\gamma$ and $\left\|y - x\right\|_1 \le 2\gamma$.

Case 1 ($\|y-x\|_1 > 2\gamma$): For all $\tilde{e}$ let $\|\tilde{e}-x\|_1 \le \gamma$, then we have $\|\tilde{e}-y\|_1 > \gamma \Rightarrow \mu(\tilde{e}-y) = 0$, so

$$\int\limits_{\|\tilde{e}-x\|_1\le\gamma} |\mu(\tilde{e}-y) - \mu(\tilde{e}-x)|\,d\tilde{e} = 1.$$

Similarly, for all $\tilde{e}$ let $\|\tilde{e}-y\|_1 \le \gamma$, then we have $\mu(\tilde{e}-x) = 0$, so

$$\int\limits_{\|\tilde{e}-y\|_1\le\gamma} |\mu(\tilde{e}-y) - \mu(\tilde{e}-x)|\,d\tilde{e} = 1.$$

Hence,

$$\int\limits_{Q_\gamma} |\mu(\tilde{e}-y) - \mu(\tilde{e}-x)|\,d\tilde{e} = \int\limits_{\|\tilde{e}-x\|_1\le\gamma} |\mu(\tilde{e}-y) - \mu(\tilde{e}-x)|\,d\tilde{e} + \int\limits_{\|\tilde{e}-y\|_1\le\gamma} |\mu(\tilde{e}-y) - \mu(\tilde{e}-x)|\,d\tilde{e} = 2.$$

Because $2 < \dfrac{\|y-x\|_1}{\gamma} \le \dfrac{d^{1-1/p}\|y-x\|_p}{\gamma}$, and considering the fact that for $p \in [1,2]$ satisfies $\|y-x\|_1 \le d^{1-\frac{1}{p}}\|y-x\|_p$, we get the following inequality:

$$\int\limits_{Q_\gamma} |\mu(\tilde{e}-y) - \mu(\tilde{e}-x)|\,d\tilde{e} \le \frac{d^{1-1/p}\|y-x\|_p}{\gamma}.$$

Case 2 ($\|y-x\|_1 \le 2\gamma$): Expand the integral I_1 into 4 integrals.

$$\begin{aligned}
&\int\limits_{Q_\gamma} |\mu(\tilde{e}-y) - \mu(\tilde{e}-x)|\,d\tilde{e}\\
&= \int\limits_{\|\tilde{e}-x\|_1\le\gamma\ \&\ \|\tilde{e}-y\|_1\le\gamma} |\mu(\tilde{e}-y) - \mu(\tilde{e}-x)|\,d\tilde{e} + \int\limits_{\|\tilde{e}-x\|_1\le\gamma\ \&\ \|\tilde{e}-y\|_1\ge\gamma} |\mu(\tilde{e}-y) - \mu(\tilde{e}-x)|\,d\tilde{e}\\
&+ \int\limits_{\|\tilde{e}-x\|_1\ge\gamma\ \&\ \|\tilde{e}-y\|_1\le\gamma} |\mu(\tilde{e}-y) - \mu(\tilde{e}-x)|\,d\tilde{e} + \int\limits_{\|\tilde{e}-x\|_1\ge\gamma\ \&\ \|\tilde{e}-y\|_1\ge\gamma} |\mu(\tilde{e}-y) - \mu(\tilde{e}-x)|\,d\tilde{e}.
\end{aligned}$$

Now let's consider each integral from the expansion separately:

1. For the first and fourth integrals, the following is true:

$$\mu(\tilde{e}-x) = \mu(\tilde{e}-y).$$

Then, substituting into the first integral, we get

$$\int\limits_{\|\tilde{e}-x\|_1\le\gamma\ \&\ \|\tilde{e}-y\|_1\le\gamma} |\mu(\tilde{e}-y) - \mu(\tilde{e}-x)|\,d\tilde{e} = 0. \tag{15}$$

Similarly, substituting into the fourth integral, we get

$$\int\limits_{\|\tilde{e}-x\|_1\ge\gamma\ \&\ \|\tilde{e}-y\|_1\ge\gamma} |\mu(\tilde{e}-y) - \mu(\tilde{e}-x)|\,d\tilde{e} = 0. \tag{16}$$

2. When $\|\tilde{e}-x\|_1 \le \gamma\ \&\ \|\tilde{e}-y\|_1 \ge \gamma$ or $\|\tilde{e}-x\|_1 \ge \gamma\ \&\ \|\tilde{e}-y\|_1 \le \gamma$, we have

$$\int\limits_{\|\tilde{e}-x\|_1\le\gamma\ \&\ \|\tilde{e}-y\|_1\ge\gamma} |\mu(\tilde{e}-y) - \mu(\tilde{e}-x)|\,d\tilde{e} = \int\limits_{\|\tilde{e}-x\|_1\ge\gamma\ \&\ \|\tilde{e}-y\|_1\le\gamma} |\mu(\tilde{e}-y) - \mu(\tilde{e}-x)|\,d\tilde{e},$$

where $\mu(\tilde{e}-x)$ and $\mu(\tilde{e}-y)$ do not intersect, therefore, using this and the symmetry of integrals, and by defining the set $S = \left\{\tilde{e} \in \mathbb{R}^d \,|\, \|\tilde{e}-x\|_1 \le \gamma \text{ and } \|\tilde{e}-y\|_1 \ge \gamma\right\}$, we get the sum of the second and third integrals

$$2\int\limits_{\|\tilde{e}-x\|_1\le\gamma\ \&\ \|\tilde{e}-y\|_1\ge\gamma} |\mu(\tilde{e}-y) - \mu(\tilde{e}-x)|\,d\tilde{e} = \frac{2}{c_d\gamma^d}V_S, \tag{17}$$

where V_S is volume on the set S.

Summing up the values of the four integrals (15)–(17), we get:

$$\int_{Q_\gamma} |\mu(\tilde{e}-y)-\mu(\tilde{e}-x)|\,d\tilde{e} = \frac{2}{c_d\gamma^d}V_S, \tag{18}$$

where V_S is the volume on the set S.

Now let's find the upper bound of V_S by $\|y-x\|_p$. Let $V_{\text{cap}}(r)$—the volume of a spherical cap with a distance of r to the center of the l_1-sphere. Then

$$V_S = c_d\gamma^d - 2V_{\text{cap}}\left(\frac{\|y-x\|_p}{4}\right). \tag{19}$$

The volume of a d-dimensional spherical cap V_{cap} can be calculated in terms of a $(d-1)$-dimensional l_1-sphere as follows:

$$V_{\text{cap}}(r) = \int_r^\gamma c_{d-1}(\gamma-\rho)^{d-1}\,d\rho \quad \text{for} \quad r\in[0,\gamma],$$

where $c_d = \dfrac{2^d}{d!}$ and $d \geq 1$. For $r\in[0,\gamma]$ we have

$$V'_{\text{cap}}(r) = -c_{d-1}(\gamma-r)^{d-1} \leq 0,$$
$$V''_{\text{cap}}(r) = (d-1)c_{d-1}(\gamma-r)^{d-2} \geq 0,$$

where V'_{cap}, V''_{cap} are the first and second derivatives with respect to r, respectively. Hence, V_{cap} is convex on $[0,\gamma]$, and by definition of the subgradient we have

$$V_{\text{cap}}(0) + V'_{\text{cap}}(0)r \leq V_{\text{cap}}(r) \quad \text{for} \quad r\in[0,\gamma].$$

Since $V_{\text{cap}}(0) = \dfrac{1}{2}c_d\gamma^d$ and $V'_{\text{cap}}(0) = -c_{d-1}\gamma^{d-1}$, it follows that

$$\frac{1}{2}c_d\gamma^d - c_{d-1}\gamma^{d-1} \leq V_{\text{cap}}(r) \quad \text{for} \quad r\in[0,\gamma]. \tag{20}$$

Because $\|y-x\|_1/2 \leq \gamma$ and noting that $\|y-x\|_p/4 \leq \|y-x\|_p/2 \leq \|y-x\|_1/2$ holds for $p\in[1,2]$, we can set $r = \|y-x\|_p/4 \leq \gamma$ and substitute in (20). By doing this and using (19) we get

$$V_S = c_d\gamma^d - 2V_{\text{cap}}\left(\frac{\|y-x\|_p}{4}\right) \leq 2c_{d-1}\gamma^{d-1}\frac{\|y-x\|_p}{4}.$$

Now substituting the obtained estimate of V_S in (18), we have

$$\int_{Q_\gamma} |\mu(\tilde{e}-y)-\mu(\tilde{e}-x)|\,d\tilde{e} \leq \frac{c_{d-1}}{c_d}\frac{\|y-x\|_p}{\gamma}.$$

Since $c_d = \dfrac{2^d}{d!}$, it can be seen that

$$\int_{Q_\gamma} |\mu(\tilde{e}-y)-\mu(\tilde{e}-x)|\,d\tilde{e} \leq \frac{d}{2\gamma}\|y-x\|_p.$$

Now, having obtained an estimate for the integral I_1, we have that

$$\|\nabla f_\gamma(y) - \nabla f_\gamma(x)\|_q \leq \frac{dM}{2\gamma}\|y-x\|_p.$$

5.2. Proof of Lemma 2

In this subsection we will focus on the proof of Lemma 2 for l_2-randomization. In many works, for example in [1, 3, 9], the estimate of the second moment for the gradient-free approximation (5) is given and proved up to a certain numerical constant c. In this proof, we will figure out what this numerical constant c is equal to. And a detailed proof of Lemma 2 for l_1-randomization is given in Lemma 4 [4].

For all $x \in Q$ with Assumptions 1 and 2 then $\nabla f_\gamma(x,\xi,e)$ of (5) has a lower bound (second moment):

$$E_{\xi,e}\left[\left\|\nabla f_\gamma(x,\xi,e)\right\|_q^2\right] \le \kappa(p,d)\left(dM_2^2 + \frac{d^2\Delta^2}{\sqrt{2}\gamma^2}\right),$$

where $1/p + 1/q = 1$ and

$$\kappa(p,d) = \sqrt{2}\min\{q, \ln d\}\, d^{2/q-1}.$$

Proof. Let's consider

$$\begin{aligned}
\mathbb{E}_{\xi,e}\left[\left\|\nabla f_\gamma(x,\xi,e)\right\|_q^2\right] &= \mathbb{E}_{\xi,e}\left[\left\|\frac{d}{2\gamma}(f_\delta(x+\gamma e,\xi) - f_\delta(x-\gamma e,\xi))e\right\|_q^2\right] \\
&= \frac{d^2}{4\gamma^2}\mathbb{E}_{\xi,e}\left[\|e\|_q^2\left(f(x+\gamma e,\xi) + \delta(x+\gamma e) - f(x-\gamma e,\xi) - \delta(x-\gamma e)\right)^2\right] \qquad (21) \\
&\le \frac{d^2}{2\gamma^2}\left(\mathbb{E}_{\xi,e}\left[\|e\|_q^2\left(f(x+\gamma e,\xi) - f(x-\gamma e,\xi)\right)^2\right] + \mathbb{E}_e\left[\|e\|_q^2\left(\delta(x+\gamma e) - \delta(x-\gamma e)\right)^2\right]\right),
\end{aligned}$$

where we used the fact that for all $a, b, (a+b)^2 \le 2a^2 + 2b^2$. For the first term (21), the following holds with an arbitrary parameter α taking into account the symmetric distribution of e

$$\begin{aligned}
&\frac{d^2}{2\gamma^2}\mathbb{E}_{\xi,e}\left[\|e\|_q^2\left(f(x+\gamma e,\xi) - f(x-\gamma e,\xi)\right)^2\right] \\
&= \frac{d^2}{2\gamma^2}\mathbb{E}_{\xi,e}\left[\|e\|_q^2\left((f(x+\gamma e,\xi) - \alpha) - (f(x-\gamma e,\xi) - \alpha)\right)^2\right] \\
&\le \frac{d^2}{\gamma^2}\mathbb{E}_{\xi,e}\left[\|e\|_q^2\left(f(x+\gamma e,\xi) - \alpha\right)^2 + \left(f(x-\gamma e,\xi) - \alpha\right)^2\right] \qquad (22) \\
&= \frac{d^2}{\gamma^2}\left(\mathbb{E}_{\xi,e}\left[\|e\|_q^2\left(f(x+\gamma e,\xi) - \alpha\right)^2\right] + \mathbb{E}_{\xi,e}\left[\left(f(x-\gamma e,\xi) - \alpha\right)^2\right]\right) \\
&= \frac{2d^2}{\gamma^2}\mathbb{E}_{\xi,e}\left[\|e\|_q^2\left(f(x+\gamma e,\xi) - \alpha\right)^2\right].
\end{aligned}$$

Applying the Cauchy–Schwartz inequality for (22) and using $\sqrt{\mathbb{E}\left[\|e\|_q^4\right]} \le \kappa'(p,d)$, where $\kappa'(p,d) = \min\{q, \ln d\}\, d^{2/q-1}$, we get

$$\begin{aligned}
\frac{2d^2}{\gamma^2}\mathbb{E}_{\xi,e}\left[\|e\|_q^2\left(f(x+\gamma e,\xi) - \alpha\right)^2\right] &\le \frac{2d^2}{\gamma^2}\mathbb{E}_\xi\left[\sqrt{\mathbb{E}\left[\|e\|_q^4\right]}\sqrt{\mathbb{E}_e\left[\left(f(x+\gamma e,\xi) - \alpha\right)^4\right]}\right] \\
&\le \frac{2d^2\kappa'(p,d)}{\gamma^2}\mathbb{E}_\xi\left[\sqrt{\mathbb{E}_e\left[\left(f(x+\gamma e,\xi) - \alpha\right)^4\right]}\right]. \qquad (23)
\end{aligned}$$

The following lemma, which we will consider, is a refinement of Lemma 9 [3] with the indication of a numerical constant c.

Lemma 4. *For any function $f(e)$, that is M-Lipschitz with respect to the l_2-norm, it holds that if e is uniformly distributed on the Euclidean sphere, then*

$$\sqrt{\mathbb{E}\left[\left(f(e) - \mathbb{E}[f(e)]\right)^4\right]} \le \frac{M_2^2}{\sqrt{2}d}.$$

Proof. Let the measure concentration inequality, which uses mathematical expectation, look like this:

$$\Pr(|f - \mathbb{E}f| > t) \le K\exp(-\eta t^2),$$

where K and η are unknown parameters. Then, in order to find the parameters K and η, we write an inequality using the median of the function using the parameters K and η (since in the literature, for example, in [10], the inequality of measure concentration on the sphere is usually written using the median of the function M_f, and not the mathematical expectation):

$$\Pr(|f - M_f| > t) \le 2K\exp(-\eta t^2/4) = 4\exp(-dt^2/2M_2^2). \tag{24}$$

Substituting the parameters $K = 2$ and $\eta = 2d/M_2^2$ from inequality (24), we write down the standard result on the concentration of Lipschitz functions on the Euclidean unit sphere

$$\Pr(|f(e) - \mathbb{E}[f(e)]| > t) \le 2\exp(-2dt^2/M_2^2).$$

Hence,

$$\sqrt{\mathbb{E}\left[(f(e) - \mathbb{E}[f(e)])^4\right]} = \sqrt{\int_{t=0}^{\infty} \Pr\left((f(e) - \mathbb{E}[f(e)])^4 > t\right)dt}$$

$$= \sqrt{\int_{t=0}^{\infty} \Pr\left(|f(e) - \mathbb{E}[f(e)]| > \sqrt[4]{t}\right)dt} \le \sqrt{\int_{t=0}^{\infty} 2\exp\left(-\frac{2d\sqrt{t}}{M_2^2}\right)dt} = \sqrt{2\frac{M_2^4}{(2d)^2}},$$

where the last step used the fact that $\int_{t=0}^{\infty}\exp(-\sqrt{x})dx = 2$. Thus, a numerical constant $c = \frac{1}{\sqrt{2}}$ from Lemma 9 [3] is found.

Then we use Lemma 4 together with the fact that $f(x + \gamma e, \xi)$ is $\gamma M_2(\xi)$-Lipschitz with respect to e in terms of the l_2-norm. Thus, for (23) and $\alpha := \mathbb{E}[f(x + \gamma e, \xi)]$ we have

$$\frac{2d^2\kappa'(p,d)}{\gamma^2}\mathbb{E}_\xi\left[\sqrt{\mathbb{E}_e\left[(f(x + \gamma e, \xi) - \alpha)^4\right]}\right] \le \frac{2d^2\kappa'(p,d)}{\gamma^2}\frac{\gamma^2\mathbb{E}\left[M_2^2(\xi)\right]}{\sqrt{2}d} = \sqrt{2}\kappa'(p,d)dM_2^2. \tag{25}$$

For the second term in (21), the following holds:

$$\frac{d^2}{2\gamma^2}\mathbb{E}_e\left[\|e\|_q^2(\delta(x + \gamma e) - \delta(x - \gamma e))^2\right] \le \frac{d^2\Delta^2}{\gamma^2}\mathbb{E}_e\left[\|e\|_q^2\right] \le \frac{\kappa'(p,d)d^2\Delta^2}{\gamma^2}. \tag{26}$$

Substituting (25) and (26) into the inequality (21) and entering the coefficient $\sqrt{2}$ into $\kappa'(p,d)$, we obtain the statement of Lemma 2 for l_2-randomization with $\kappa(p,d) = \sqrt{2}\kappa'(p,d) = \sqrt{2}\min\{q, \ln d\}d^{2/q-1}$.

5.3. Proof of Lemma 3

In this subsection, we consider a brief proof of Lemma 3 for the case with l_1-randomization. The Euclidean case can be found in the following works [1, 17].

For all $x \in Q$ with Assumptions 2 and 3, and from (8), $\nabla f_\gamma(x, \xi, e)$ has the lower bound (second moment):

$$\mathbb{E}_e\left[\|\nabla f_\gamma(x, \xi, e)\|_q^2\right] \le \frac{d^{4-\frac{2}{p}}}{\gamma^2}\left(G^2 + \Delta^2\right),$$

where $1/p + 1/q = 1$.

Proof. The proof of this Lemma will be based on the proof of Lemma 4 from [4]. Using the definition of $\nabla f_\gamma(x, \xi, e)$, we get:

$$\mathbb{E}\left[\|\nabla f_\gamma(x, \xi, e)\|_q^2\right] = \frac{d^2}{\gamma^2}\mathbb{E}\left[(f(x + \gamma e) + \delta(x))^2\|\mathrm{sgn}(e)\|_q^2\right] = \frac{d^{4-\frac{2}{p}}}{\gamma^2}\mathbb{E}\left[(f(x + \gamma e) + \delta(x))^2\right].$$

Further, with Assumptions 2 and 3, we obtain the statement of the lemma:

$$E\left[\left\|\nabla f_\gamma(x,\xi,e)\right\|_q^2\right] = \frac{d^{4-\frac{2}{p}}}{\gamma^2} E\left[(f(x+\gamma e)+\delta(x))^2\right] \le \frac{d^{4-\frac{2}{p}}}{\gamma^2}\left(G^2+\Delta^2\right).$$

5.4. Estimates of the Level of Noise

In this subsection, we present the necessary lemmas and corollaries for finding two estimates of the level of noise (adversarial noise): for l_1-randomization (4) and for l_2-randomization (5). To do this, we use the known results and the same assertion as in [18].

Lemma 5 (see [4]). *The function $f_\gamma(x)$ is differentiable with the following l_1-stochastic gradient:*

$$\nabla f_\gamma(x) = \mathbb{E}_e\left[\frac{d}{\gamma} f(x+\gamma e)\operatorname{sgn}(e)\right].$$

Lemma 6 (see [19]). *The function $f_\gamma(x)$ is differentiable with the following l_2-stochastic gradient:*

$$\nabla f_\gamma(x) = \mathbb{E}_e\left[\frac{d}{\gamma} f(x+\gamma e)e\right].$$

Lemma 7 (see [20]). *Let the vector e be a random unit vector from the Euclidean unit sphere $\{e : \|e\|_2 = 1\}$. Then for all $r \in \mathbb{R}^d$ it follows*

$$\mathbb{E}\left[|\langle e, r\rangle|\right] \le \frac{\|r\|_2}{\sqrt{d}}.$$

Lemma 8. *For $\nabla f_\gamma(x,\xi,e)$ and $\nabla f_\gamma(x)$ with Assumption 2, the following holds:*

$$\mathbb{E}_{\xi,e}[\langle \nabla f_\gamma(x,\xi,e), r\rangle] \ge \langle \nabla f_\gamma(x), r\rangle - \frac{d\Delta\mathbb{E}_e\left[|\langle \operatorname{sgn}(e), r\rangle|\right]}{\gamma},$$

where ∇f_γ with l_1-randomization;

$$\mathbb{E}_{\xi,e}[\langle \nabla f_\gamma(x,\xi,e), r\rangle] \ge \langle \nabla f_\gamma(x), r\rangle - \frac{d\Delta\mathbb{E}_e[|\langle e, r\rangle|]}{\gamma},$$

where ∇f_γ with l_2-randomization.

Proof. Consider the following statements.

(i) for l_1-randomization (4):

$$\begin{aligned}\nabla f_\gamma(x,\xi,e) &= \frac{d}{2\gamma}(f_\delta(x+\gamma e,\xi) - f_\delta(x-\gamma e,\xi))\operatorname{sgn}(e)\\ &= \frac{d}{2\gamma}(f(x+\gamma e,\xi)+\delta(x+\gamma e) - f(x-\gamma e,\xi)-\delta(x-\gamma e))\operatorname{sgn}(e)\\ &= \frac{d}{2\gamma}((f(x+\gamma e,\xi) - f(x-\gamma e,\xi))\operatorname{sgn}(e) + (\delta(x+\gamma e)-\delta(x-\gamma e))\operatorname{sgn}(e)).\end{aligned}$$

From this equality it follows that

$$\begin{aligned}\mathbb{E}_{\xi,e}[\langle \nabla f_\gamma(x,\xi,e), r\rangle] &= \frac{d}{2\gamma} E_{\xi,e}[\langle (f(x+\gamma e,\xi) - f(x-\gamma e,\xi))\operatorname{sgn}(e), r\rangle]\\ &+ \frac{d}{2\gamma}\mathbb{E}_e[\langle (\delta(x+\gamma e)-\delta(x-\gamma e))\operatorname{sgn}(e), r\rangle].\end{aligned} \tag{27}$$

Applying Lemma 5 to the first term (27), we obtain

$$\begin{aligned}&\frac{d}{2\gamma}\mathbb{E}_{\xi,e}[\langle(f(x+\gamma e,\xi)-f(x-\gamma e,\xi))\operatorname{sgn}(e),r\rangle]\\&=\frac{d}{2\gamma}\mathbb{E}_{\xi,e}[\langle f(x+\gamma e,\xi)\operatorname{sgn}(e),r\rangle]+\frac{d}{2\gamma}\mathbb{E}_{\xi,e}[\langle f(x-\gamma e,\xi)\operatorname{sgn}(e),r\rangle]\\&=\frac{d}{\gamma}\mathbb{E}_e\left[\langle\mathbb{E}_\xi\left[f(x+\gamma e,\xi)\right]\operatorname{sgn}(e),r\rangle\right]=\frac{d}{\gamma}\mathbb{E}_e[\langle f(x+\gamma e)\operatorname{sgn}(e),r\rangle]=\langle\nabla f_\gamma(x),r\rangle.\end{aligned}\tag{28}$$

For the second term (27), taking into account $|\delta(x)|\le\Delta$ we get

$$\frac{d}{\gamma}\mathbb{E}_e[\langle(\delta(x+\gamma e)-\delta(x-\gamma e))\operatorname{sgn}(e),r\rangle]\ge-\frac{d}{2\gamma}2\Delta\mathbb{E}_e[|\langle\operatorname{sgn}(e),r\rangle|]=-\frac{d}{\gamma}\Delta\mathbb{E}_e[|\langle\operatorname{sgn}(e),r\rangle|].\tag{29}$$

Using equations (28) and (29) for equation (27), we obtain a statement of the lemma for l_1-randomization.

(ii) for l_2-randomization (5):

$$\begin{aligned}\nabla f_\gamma(x,\xi,e)&=\frac{d}{2\gamma}(f_\delta(x+\gamma e,\xi)-f_\delta(x-\gamma e,\xi))e\\&=\frac{d}{2\gamma}(f(x+\gamma e,\xi)+\delta(x+\gamma e)-f(x-\gamma e,\xi)-\delta(x-\gamma e))e\\&=\frac{d}{2\gamma}((f(x+\gamma e,\xi)-f(x-\gamma e,\xi))e+(\delta(x+\gamma e)-\delta(x-\gamma e))e).\end{aligned}$$

From this equality it follows that

$$\begin{aligned}E_{\xi,e}[\langle\nabla f_\gamma(x,\xi,e),r\rangle]&=\frac{d}{2\gamma}\mathbb{E}_{\xi,e}[\langle(f(x+\gamma e,\xi)-f(x-\gamma e,\xi))e,r\rangle]\\&+\frac{d}{2\gamma}\mathbb{E}_e[\langle(\delta(x+\gamma e)-\delta(x-\gamma e))e,r\rangle].\end{aligned}\tag{30}$$

Applying Lemma 6 to the first term (30), we obtain

$$\begin{aligned}&\frac{d}{2\gamma}\mathbb{E}_{\xi,e}[\langle(f(x+\gamma e,\xi)-f(x-\gamma e,\xi))e,r\rangle]\\&=\frac{d}{2\gamma}\mathbb{E}_{\xi,e}[\langle f(x+\gamma e,\xi)e,r\rangle]+\frac{d}{2\gamma}\mathbb{E}_{\xi,e}[\langle f(x-\gamma e,\xi)e,r\rangle]\\&=\frac{d}{\gamma}\mathbb{E}_e\left[\langle\mathbb{E}_\xi\left[f(x+\gamma e,\xi)\right]e,r\rangle\right]=\frac{d}{\gamma}\mathbb{E}_e[\langle f(x+\gamma e)e,r\rangle]=\langle\nabla f_\gamma(x),r\rangle.\end{aligned}\tag{31}$$

For the second term (30), taking into account that $|\delta(x)|\le\Delta$ we obtain

$$\frac{d}{\gamma}\mathbb{E}_e[\langle(\delta(x+\gamma e)-\delta(x-\gamma e))e,r\rangle]\ge-\frac{d}{2\gamma}2\Delta\mathbb{E}_e[|\langle e,r\rangle|]=-\frac{d}{\gamma}\Delta\mathbb{E}_e[|\langle e,r\rangle|].\tag{32}$$

Using equations (31) and (32) for equation (30), we obtain a statement of the lemma for l_2-randomization.

Corollary 5. Note that the vector $\operatorname{sgn}(e)=\left(\operatorname{sgn}(e_1),\ldots,\operatorname{sgn}(e_d)\right)$ in the l_1-norm behaves in a similar way for a large space dimension d as a vector of independent Rademacher random variables. Then Khinchin's inequality implies $E[|\langle\operatorname{sgn}(e),r\rangle|]\le\|r\|_2$, where $r\in\mathbb{R}^d$. Applying this inequality to Lemma 8, we obtain

$$\mathbb{E}_e\langle[\nabla f_\gamma(x,e)]-\nabla f_\gamma(x),r\rangle\lesssim\frac{d\Delta\|r\|_2}{\gamma}.$$

Corollary 6. Applying the assertion of Lemma 7 to Lemma 8, we obtain the same inequality as in [18] for all $r\in\mathbb{R}^d$

$$\mathbb{E}_e\langle[\nabla f_\gamma(x,e)]-\nabla f_\gamma(x),r\rangle\lesssim\frac{\sqrt{d}\Delta\|r\|_2}{\gamma}.$$

Now consider how to get estimates of the level of noise.

From (13) and (14), substituting the value of the parameter γ from Corollary 1 ($\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ for l_1-randomization, $\gamma = \frac{\varepsilon}{2M_2}$ for l_2-randomization) and $R = \left\|x^0 - x_*\right\|_2$, we obtain bounds on the level of noise for l_1 and l_2-randomization:

Smoothing scheme with l_1-randomization	Smoothing scheme with l_2-randomization
$\Delta \lesssim \frac{\varepsilon^2}{\sqrt{d}M_2R}$	$\Delta \lesssim \frac{\varepsilon^2}{\sqrt{d}M_2R}$

Remark 4. It is worth noting that the results of Lemma 8 and Corollaries 5 and 6 for saddle-point problems are the same and are proved in a similar way. Therefore, the estimates of the level of noise for the problem with a saddle-point coincide with the estimates for convex optimization. In order to avoid repetition, we will refer to this remark as the result of obtaining estimates of the level of noise for saddle-point problems.

6. NUMERICAL EXPERIMENTS

In this section, we present a numerical comparison of two smoothing techniques in a federated learning architecture. We consider a stochastic non-smooth optimization problem on the simplex set $Q = \left\{x \in \mathbb{R}^d : \|x\|_1 = 1, x \geq 0\right\}$ with function $f : \mathbb{R}^d \to \mathbb{R}$, defined as follows:

$$f(x) = \langle b, x\rangle + \|x\|_\infty,$$

where $b \in \mathbb{R}^d$ – a random vector uniformly distributed on the interval $[0,1]$. The gradient-free two-point algorithm Minibatch Accelerated SGD, discussed in Section 4, is used as an optimization method. Figure 2 shows the dependence of the error $f(\bar{x})$ at the last iteration on the change in the number of local calls of the gradient-free oracle $K = 3^0,\ldots,3^8$ with a different number of machines $B = 8$ and $B = 32$, working in parallel, where $\bar{x} = \frac{1}{t}\sum_{i=1}^{t} x_i$, and t is the iteration number. The number of iterations was chosen as $KN = 3^8$. The larger K, the smaller the number of communication rounds N, and the converse is also true. The level of noise $\Delta = 0$ (without adversarial noise), and the dimension of the problem $d = 100$.

It is easy to see from Fig. 2 that as the number of local updates K increases (a decrease in communication rounds N), the accuracy get worse (thereby confirming the theoretical results), but it is not critical. That is, when solving practical problems, despite the theory, you can take the number of local updates $K \leq 3^6$, to get a good enough result. It is also worth noting that, in practice, the smoothing scheme with l_2-randomization is not only matches the smoothing scheme with l_1-randomization, but sometimes surpasses it. However, it is interesting to find out in which cases the smoothing scheme with l_1-randomization will be superior to the smoothing scheme with l_2-randomization. To do this, consider two cases of com-

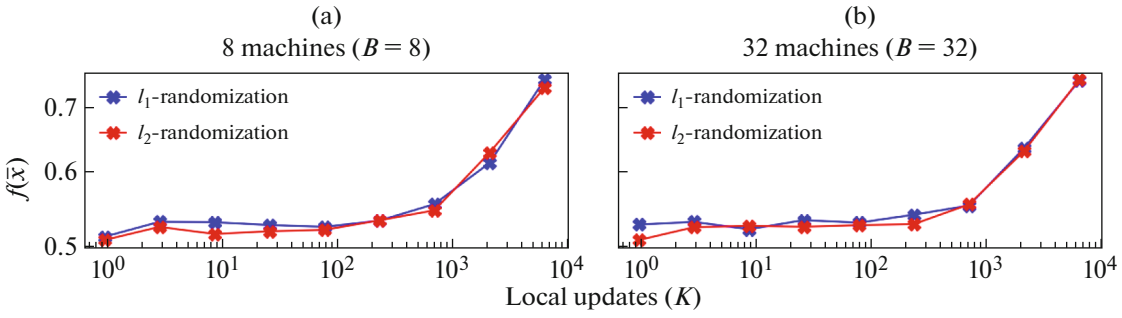

Fig. 2. The dependence of the error on the number of machines B and various local updates K.

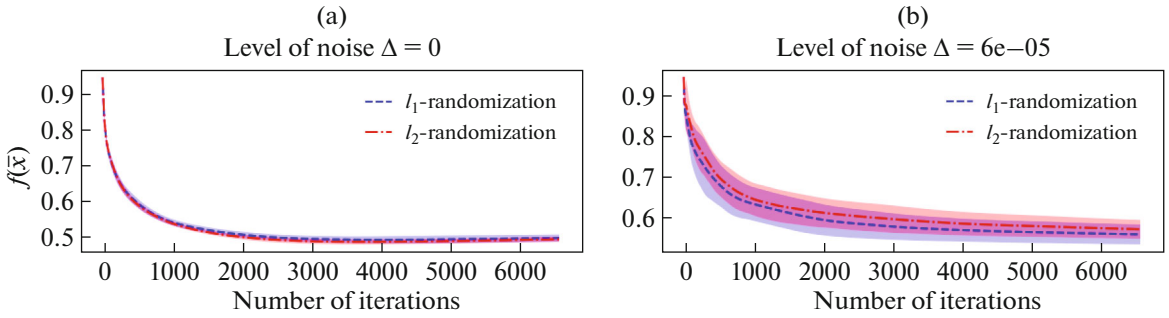

Fig. 3. The dependence of the error on the number of iterations at different values of the inaccuracy level at 20 runs.

puting a gradient-free oracle: with adversarial noise and without adversarial noise. Figure 3 shows the dependence of the error $f(\bar{x})$ on the number of iterations and the level of noise (presence of adversarial noise) $\Delta = 0$ and $\Delta = 6\times10^{-5}$. The number of machines working in parallel is chosen to be $B = 8$. On each machine, the number of local calls to the gradient-free oracle is $K = 3^2$, and the number of communication rounds 3^6. Thus, the total number of iterations is $NK = 3^8$. The dimension of the problem $d = 100$, the number of launches is 20.

According to Fig. 3, it can be concluded that when adversarial noise is added, the smoothing scheme with l_1 randomization works better than the smoothing scheme with l_2 randomization in the architecture of federated learning.

7. CONCLUSIONS

In this paper, we obtained upper bounds for the optimal algorithm for solving saddle-point problems in the federated learning architecture and found the Lipschitz constant for the gradient in the smoothing scheme with l_1-randomization. Using smoothing schemes, we have created optimal gradientl-free two-point and one-point algorithms with l_1 and l_2-randomization, thanks to which it is possible to solve stochastic non-smooth convex optimization problems and convex-concave optimization problems in the federated learning setting with a stochastic gradient-free oracle. We showed under what conditions the smoothing scheme with l_1-randomization works better than with l_2-randomization in the federated learning architecture. It has been shown in practice that the number of local updates can be increased by reducing the number of communication rounds, while the total number of iterations remains the same.

APPENDIX

A. UPPER BOUNDS

Consider a smooth convex-concave saddle-point problem

$$\min_{x\in Q_x\subseteq\mathbb{R}^{d_x}}\max_{y\in Q_y\subseteq\mathbb{R}^{d_y}} f(x,y), \tag{A.33}$$

where $f: Q_x\times Q_y\to\mathbb{R}$ is a convex-concave Lipschitz continuous function, Q_x and Q_y are convex sets. For simplicity of presentation, we introduce the set $Q_z = Q_x\times Q_y$, $z = (x,y)$ and the monotone operator F:

$$F(z) = F(x,y) = \begin{pmatrix}\nabla_x f(x,y)\\ -\nabla_y f(x,y)\end{pmatrix}. \tag{A.34}$$

Then $z_*\in Q_z$—the solution of the variational inequality (VI), looks as follows:

$$\langle F(z), z_* - z\rangle \le 0 \quad \forall z\in Q_z. \tag{A.35}$$

We estimate the inaccuracy of a possible solution $z\in Q_z$ by the error

$$\mathrm{Err}_{\mathrm{vi}}(z) = \max_{u\in Q_z}\langle F(u), z-u\rangle. \tag{A.36}$$

In what follows, we impose on F, in addition to being monotonic, the requirement

$$\forall(z,z'\in Q_z):\|F(z)-F(z')\|_*\le L\|z-z'\|+V, \tag{A.37}$$

where $L\ge 0$, $V\ge 0$ are known constants, $\|\cdot\|_*=\max_{z:\|z\|\le 1}\langle\cdot,z\rangle$. And we assume that $F(z,\xi_t)$ is unbiased and has bounded variance, i.e., $\forall z\in Q_z$ we have

$$\mathbb{E}[F(z,\xi_t)]=F(z),\quad \mathbb{E}[\|F(z,\xi_t)-F(z)\|_*^2]\le\sigma^2.$$

To extend the main result of [11] to saddle-point problems and VI, we choose Stochastic Mirror Prox (SMP) described in [21] as an algorithm. In the future, we will consider two methods of the FL architecture proposed in [11] and call them Minibatch SMP and Single-Machine SMP. To do this, we use the well-known results and assumptions.

Algorithm 1 SMP

1: Initialization. Select $r_0\in Z^0$ and step size $\eta_\tau, 1\le\tau\le t$.

2: Step τ, $\tau=1,2,\ldots,t$: According to the known $r_{\tau-1}\in Z^0$, calculate

$$w_\tau=P_{r_{\tau-1}}(\eta_\tau\hat F(r_{\tau-1})),\quad r_\tau=P_{r_{\tau-1}}(\eta_\tau\hat F(w_\tau)). \tag{A.38}$$

When $\tau<t$, execute the loop before the step $t+1$.

3: At the step t output

$$\hat z_t=\left[\sum_{\tau=1}^t\eta_\tau\right]^{-1}\sum_{\tau=1}^t\eta_\tau w_\tau.$$

Assumption 4. For every $z\in Q_z$ with $\mu\in[0,\infty)$ we have

$$\|\mathbb{E}[F(z,\xi_t)-F(z)]\|_*\le\mu,\quad \mathbb{E}[\|F(z,\xi_t)-F(z)\|_*^2]\le\sigma^2.$$

Assumption 5. For every $z\in Q_z$ and for every t we have

$$\mathbb{E}\left[\exp\left\{\|F(z,\xi_t)-F(z)\|_*^2/\sigma^2\right\}\right]\le\exp\{1\}.$$

Lemma 9 (see [21]). *Let VI* (A.35) *with a monotone operator F* (A.34)*, satisfying requirement* (A.37) *be solved using a t step algorithm* 8*, using a stochastic oracle ($\hat F=F(z,\xi_t)$), and let the step sizes $\eta_t\equiv\eta$ satisfy $0\le\eta\le\frac{1}{\sqrt{3}L}$. Then*

(i) *with Assumption* 4 *we have*

$$\mathbb{E}\{\mathrm{Err}_{\mathrm{vi}}(\hat z_t)\}\le K_0(t)\equiv\left[\frac{R^2}{t\eta}+\frac{7\eta}{2}\left[V^2+2\sigma^2\right]\right]+2\mu R;$$

(ii) *with Assumptions* 4, 5 *for any $\Lambda>0$*

$$\mathrm{Prob}\{\mathrm{Err}_{\mathrm{vi}}(\hat z_t)>K_0(t)+\Lambda K_1(t)\}\le\exp\left\{-\Lambda^2/3\right\}+\exp\{-\Lambda t\},$$

where

$$K_1(t)=\frac{7\sigma^2\eta}{2}+\frac{2\sigma R}{\sqrt{t}}.$$

Corollary 7. Using the results of Lemma 9, we extend the idea of the optimal algorithm [11] and obtain upper bounds.

• **Minibatch SMP**

This algorithm performs N iterations of the SMP using mini-batch gradients of size BK. During each round of communication, each machine calculates K stochastic oracle, then the machines send their mini-batches, averaging into one large mini-batch size BK, then they update w_τ and r_τ in accordance with

(A.38). Since the calculation of the processed stochastic oracle reduces the variance by a factor of BK, then we denote $\sigma_{BK}^2 = \frac{\sigma^2}{BK}$. Assume that the step size $\eta_t \equiv \eta$ of algorithm 8 is

$$\eta_t = \min\left[\frac{1}{\sqrt{3}L}, 7R\sqrt{\frac{2BK}{7N(V^2 + 2\sigma^2)}}\right], \tag{A.39}$$

where N is the given number of iterations. Then with Assumption 4

$$\mathbb{E}\{\mathrm{Err}_{\mathrm{vi}}(\hat{z}_N)\} \le K_0^*(N) \equiv \max\left[\frac{7}{4}\frac{LR^2}{N}, 7R\sqrt{\frac{V^2 + 2\sigma^2}{3BKN}}\right] + 2\mu R. \tag{A.40}$$

Since the upper bound (A.40) for the error of algorithm 8 with the step strategy (A.39) depends in a similar way on σ and V, we can replace V with σ. Then

$$\mathbb{E}\{\mathrm{Err}_{\mathrm{vi}}(\hat{z}_N)\} \le K_0^*(N) \equiv \max\left[\frac{7}{4}\frac{LR^2}{N}, 7\frac{\sigma R}{\sqrt{BKN}}\right] + 2\mu R,$$

if Assumptions 4 and 5 hold, then

$$\mathrm{Prob}\left\{\mathrm{Err}_{\mathrm{vi}}(\hat{z}_N) > K_0^*(N) + \Lambda K_1^*(N)\right\} \le \exp\left\{-\Lambda^2/3\right\} + \exp\{-\Lambda N\},$$

where

$$K_1^*(N) = \frac{7}{2}\frac{\sigma_{BK}R}{\sqrt{N}} = \frac{7}{2}\frac{\sigma R}{\sqrt{BKN}}.$$

- **Single-Machine SMP**

This algorithm, unlike Mini-batch SMP, ignores $B-1$ machines and performs KN steps of the SMP algorithm. Then assume that the step size η_t of the SMP algorithm is

$$\eta_t = \min\left[\frac{1}{\sqrt{3}L}, 7R\sqrt{\frac{2}{7KN(V^2 + 2\sigma^2)}}\right], \tag{A.41}$$

where KN is the specified number of iterations. Then with Assumption 4,

$$\mathbb{E}\{\mathrm{Err}_{\mathrm{vi}}(\hat{z}_{KN})\} \le K_0^*(KN) \equiv \max\left[\frac{7}{4}\frac{LR^2}{KN}, 7R\sqrt{\frac{V^2 + 2\sigma^2}{3KN}}\right] + 2\mu R. \tag{A.42}$$

Since the upper bound (A.42) for the error of algorithm 8 with the step strategy (A.41) depends in a similar way on σ and V, we can replace V with σ. Then

$$\mathbb{E}\{\mathrm{Err}_{\mathrm{vi}}(\hat{z}_{KN})\} \le K_0^*(KN) \equiv \max\left[\frac{7}{4}\frac{LR^2}{KN}, 7\frac{\sigma R}{\sqrt{KN}}\right] + 2\mu R,$$

if Assumptions 4 and 5 hold, then

$$\mathrm{Prob}\left\{\mathrm{Err}_{\mathrm{vi}}(\hat{z}_{KN}) > K_0^*(KN) + \Lambda K_1^*(KN)\right\} \le \exp\left\{-\Lambda^2/3\right\} + \exp\{-\Lambda KN\},$$

where

$$K_1^*(KN) = \frac{7}{2}\frac{\sigma R}{\sqrt{KN}}.$$

Let's write a lemma using the same approach as Woodworth [11] to combine two algorithms (Mini-batch SMP and Single-Machine SMP) into one optimal algorithm.

Corollary 8. For any L, R, σ, K, N, $M > 0$ the algorithm uses Mini-batch SMP when $K \le \frac{\sigma^2 N}{L^2 R^2}$, and uses Single-Machine SMP when $K > \frac{\sigma^2 N}{L^2 R^2}$, then with Assumption 4 we have

$$\mathbb{E}\{\mathrm{Err}_{\mathrm{vi}}(\hat{z})\} \le c\left(\frac{LR^2}{KN} + \frac{\sigma R}{\sqrt{BKN}} + \min\left[\frac{LR^2}{N}, \frac{\sigma R}{\sqrt{KN}}\right]\right) + 2\mu R,$$

where c is some numeric constant.

B. PROOF OF THEOREM 1

Here is a complete proof of Theorem 1. To do this, we divide the proof into two parts: the proof for l_1-randomization and the proof for l_2-randomization.

For l_1-randomization, we have the following algorithms:

• **Minibatch Accelerated SGD**

This algorithm, after N rounds of communication gives the rate of convergence for $f_\gamma(x)$ (see [11, 22]) in accordance with Corollary 5:

$$\mathbb{E}[f_\gamma(x_{ag}^{N+1}) - f(x_*)] \le \frac{4L_{f_\gamma}R^2}{N^2} + \frac{4\sigma R}{\sqrt{BKN}} + \frac{d\Delta R}{\gamma},$$

where $x_*(\gamma) = \underset{x\in Q_\gamma}{\operatorname{argmin}} f_\gamma(x)$.

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(x)$ with $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(x)$:

$$f(x_{ag}^{N+1}) - f(x_*) \le f(x_{ag}^{N+1}) - f(x_*(\gamma)) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) + \frac{2}{\sqrt{d}}\gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(x)$ you need

$$\frac{d\Delta R}{\gamma} \le \frac{\varepsilon}{6}, \tag{A.43}$$

$$\frac{4L_{f_\gamma}R^2}{N^2} \le \frac{\varepsilon}{6}, \tag{A.44}$$

$$\frac{4\sigma R}{\sqrt{BKN}} \le \frac{\varepsilon}{6}. \tag{A.45}$$

Substituting $L_{f_\gamma} = \frac{dM}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ and $\sigma^2 = 2\kappa(p,d)M_2^2$ (from Corollary 3) into inequalities (A.43)–(A.45) we get:

$$\Delta \le \frac{\gamma\varepsilon}{6dR} \Rightarrow \Delta \le \frac{\varepsilon^2}{24\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise,

$$N^2 \ge \frac{96\sqrt{d}MM_2R^2}{\varepsilon^2} \Rightarrow N \ge \frac{4\sqrt{6}d^{1/4}\sqrt{MM_2}R}{\varepsilon} \Rightarrow N = O\left(\frac{d^{1/4}\sqrt{MM_2}R}{\varepsilon}\right),$$

number of communication rounds,

$$B \ge \frac{576\sigma^2R^2}{KN\varepsilon^2} \Rightarrow B \ge \frac{1152\kappa(p,d)M_2^2R^2}{KN\varepsilon^2} \Rightarrow B = O\left(\frac{\kappa(p,d)M_2^2R^2}{KN\varepsilon^2}\right),$$

number of machines running in parallel, and

$$T = NKB = \frac{1152\kappa(p,d)M_2^2R^2}{\varepsilon^2}$$

$$\Rightarrow T = O\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) \quad \text{follows: } = \begin{cases} O\left(\dfrac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2 \quad (q = 2), \\ O\left(\dfrac{M_2^2R^2}{\varepsilon^2}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a two-point gradient-free oracle.

• **Single-Machine Accelerated SGD**

This algorithm, after NK rounds of communication gives the rate of convergence for $f_\gamma(x)$ (see [11, 22]) in accordance with Corollary 5:

$$\mathbb{E}[f_\gamma(x_{ag}^{NK+1}) - f(x_*)] \le \frac{4L_{f_\gamma}R^2}{N^2K^2} + \frac{4\sigma R}{\sqrt{NK}} + \frac{d\Delta R}{\gamma},$$

where $x_*(\gamma) = \underset{x\in Q_\gamma}{\operatorname{argmin}} f_\gamma(x)$.

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(x)$ with $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(x)$:

$$f(x_{ag}^{N+1}) - f(x_*) \le f(x_{ag}^{N+1}) - f(x_*(\gamma)) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) + \frac{2}{\sqrt{d}}\gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(x)$ you need

$$\frac{d\Delta R}{\gamma} \le \frac{\varepsilon}{6}, \tag{A.46}$$

$$\frac{4L_{f_\gamma}R^2}{K^2N^2} \le \frac{\varepsilon}{6}, \tag{A.47}$$

$$\frac{4\sigma R}{\sqrt{KN}} \le \frac{\varepsilon}{6}. \tag{A.48}$$

Substituting $L_{f_\gamma} = \frac{dM}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ and $\sigma^2 = 2\kappa(p,d)M_2^2$ (from Corollary 3), into inequalities (A.46)–(A.48), we get:

$$\Delta \le \frac{\gamma\varepsilon}{6dR} \Rightarrow \Delta \le \frac{\varepsilon^2}{24\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise and

$$NK \ge \frac{576\sigma^2R^2}{\varepsilon^2} \Rightarrow NK \ge \frac{1152\kappa(p,d)M_2^2R^2}{\varepsilon^2}.$$

Since N directly depends on K, the number of communication rounds can be taken $N = 1$, then we get:

$$NK = K = O\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right)$$

number of local calls of the gradient-free oracle and

$$T = NKB = \frac{1152\kappa(p,d)M_2^2R^2}{\varepsilon^2} \Rightarrow T = O\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} O\left(\dfrac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2 \quad (q = 2), \\ O\left(\dfrac{M_2^2R^2}{\varepsilon^2}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a two-point gradient-free oracle.

• **Local-AC-SA**

This algorithm, after N communication rounds, gives the convergence rate for $f_\gamma(x)$ (see [12, 22]) in accordance with Corollary 5:

$$\mathbb{E}[f_\gamma(x_{ag}^{N+1}) - f(x_*)] \le \frac{4L_{f_\gamma}R^2}{K^2N^2} + \frac{4\sigma R}{\sqrt{BKN}} + \frac{d\Delta R}{\gamma},$$

where $x_*(\gamma) = \underset{x\in Q_\gamma}{\operatorname{argmin}} f_\gamma(x)$.

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(x)$ with $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(x)$:

$$f(x_{ag}^{N+1}) - f(x_*) \le f(x_{ag}^{N+1}) - f(x_*(\gamma)) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) + \frac{2}{\sqrt{d}}\gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(x)$ you need

$$\frac{d\Delta R}{\gamma} \le \frac{\varepsilon}{6}, \tag{A.49}$$

$$\frac{4L_{f_\gamma}R^2}{K^2N^2} \le \frac{\varepsilon}{6}, \tag{A.50}$$

$$\frac{4\sigma R}{\sqrt{BKN}} \le \frac{\varepsilon}{6}. \tag{A.51}$$

Substituting $L_{f_\gamma} = \frac{dM}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ and $\sigma^2 = 2\kappa(p,d)M_2^2$ (from Corollary 3), into inequalities (A.49)–(A.51), we get:

$$\Delta \le \frac{\gamma\varepsilon}{6dR} \Rightarrow \Delta \le \frac{\varepsilon^2}{24\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise and

$$N^2K^2 \ge \frac{96\sqrt{d}MM_2R^2}{\varepsilon^2} \Rightarrow NK \ge \frac{4\sqrt{6}d^{1/4}\sqrt{MM_2}R}{\varepsilon}.$$

Since N directly depends on K, the number of communication rounds can be taken $N = 1$, then we get:

$$NK = K = O\left(\frac{d^{1/4}\sqrt{MM_2}R}{\varepsilon}\right),$$

the number of local calls of the gradient-free oracle,

$$B \ge \frac{576\sigma^2R^2}{KN\varepsilon^2} \Rightarrow B \ge \frac{1152\kappa(p,d)M_2^2R^2}{KN\varepsilon^2} \Rightarrow B = O\left(\frac{\kappa(p,d)M_2^2R^2}{KN\varepsilon^2}\right),$$

number of machines running in parallel and

$$T = NKB = \frac{1152\kappa(p,d)M_2^2R^2}{\varepsilon^2} \Rightarrow T = O\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} O\left(\dfrac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2 \quad (q = 2), \\ O\left(\dfrac{M_2^2R^2}{\varepsilon^2}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a two-point gradient-free oracle.

• **Federated Accelerated SGD (FedAc)**

This algorithm, after N communication rounds, gives the convergence rate for $f_\gamma(x)$ (see [13]) in accordance with Corollary 5:

$$\mathbb{E}[f_\gamma(x_{ag}^{N+1}) - f(x_*)] \le \frac{L_{f_\gamma}R^2}{KN^2} + \frac{\sigma R}{\sqrt{BKN}} + \min\left\{\frac{L_{f_\gamma}^{1/3}\sigma^{2/3}R^{4/3}}{K^{1/3}N}, \frac{L_{f_\gamma}^{1/2}\sigma^{1/2}R^{3/2}}{K^{1/4}N}\right\} + \frac{d\Delta R}{\gamma},$$

where $x_*(\gamma) = \underset{x\in Q_\gamma}{\operatorname{argmin}} f_\gamma(x)$.

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(x)$ with $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(x)$:

$$f(x_{ag}^{N+1}) - f(x_*) \le f(x_{ag}^{N+1}) - f(x_*(\gamma)) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) + \frac{2}{\sqrt{d}}\gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(x)$ you need

$$\frac{d\Delta R}{\gamma} \le \frac{\varepsilon}{8}, \tag{A.52}$$

$$\frac{L_{f_\gamma}R^2}{KN^2} \le \frac{\varepsilon}{8}, \tag{A.53}$$

$$\frac{\sigma R}{\sqrt{BKN}} \le \frac{\varepsilon}{8}, \tag{A.54}$$

$$\min\left\{\frac{L_{f_\gamma}^{1/3}\sigma^{2/3}R^{4/3}}{K^{1/3}N}, \frac{L_{f_\gamma}^{1/2}\sigma^{1/2}R^{3/2}}{K^{1/4}N}\right\} \le \frac{\varepsilon}{8}. \tag{A.55}$$

Substituting $L_{f_\gamma} = \frac{dM}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ and $\sigma^2 = 2\kappa(p,d)M_2^2$ (from Corollary 3), into inequalities (A.52)–(A.55), we get

$$\Delta \le \frac{\gamma\varepsilon}{6dR} \Rightarrow \Delta \le \frac{\varepsilon^2}{24\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise,

$$NK \ge \frac{8^{1/2}K^{1/2}L_{f_\gamma}^{1/2}R}{\varepsilon^{1/2}} \Rightarrow NK \ge \frac{16^{1/2}K^{1/2}d^{1/4}M^{1/2}M_2^{1/2}R}{\varepsilon} \Rightarrow NK \gtrsim \frac{K^{1/2}}{\varepsilon},$$

$$NK \ge \frac{64\sigma^2R^2}{B\varepsilon^2} \Rightarrow NK \ge \frac{72\kappa(p,d)dM_2^2R^2}{B\varepsilon^2} \Rightarrow NK \gtrsim \frac{1}{B\varepsilon^2},$$

and

$$NK \ge \min\left\{\frac{K^{2/3}L_{f_\gamma}^{1/3}\sigma^{2/3}R^{4/3}}{\varepsilon}, \frac{K^{3/4}L_{f_\gamma}^{1/2}\sigma^{1/2}R^{3/2}}{\varepsilon}\right\}$$

$$\Rightarrow NK \geq \min\left\{\frac{2^{1/3}K^{2/3}d^{1/6}(MM_2)^{1/3}\sigma^{2/3}R^{4/3}}{\varepsilon^{4/3}}, \frac{2^{1/2}K^{3/4}d^{1/4}(MM_2)^{1/2}\sigma^{1/2}R^{3/2}}{\varepsilon^{3/2}}\right\}$$

$$\Rightarrow NK \geq \frac{2^{1/3}K^{2/3}d^{1/6}(MM_2)^{1/3}\sigma^{2/3}R^{4/3}}{\varepsilon^{4/3}} \Rightarrow NK \gtrsim \frac{K^{2/3}}{\varepsilon^{4/3}}.$$

Thus, the smallest number of communication rounds, provided that $NK \in [1, \varepsilon^{-2}]$ and $T \in [1, \varepsilon^{-2}]$, will have the form

$$N \sim \frac{1}{\varepsilon}: \quad N \geq \frac{8L_{f_\gamma}^{1/3}\sigma^{2/3}R^{4/3}}{K^{1/3}\varepsilon} \Rightarrow N = O\left(\frac{d^{1/6}(\kappa(p,d)M)^{1/3}M_2R^{4/3}}{K^{1/3}\varepsilon^{4/3}}\right),$$

then we get:

$$K \sim \frac{1}{\varepsilon}: \quad K \geq \frac{\sigma^2R^2}{BN\varepsilon^2} \Rightarrow K = O\left(\frac{\kappa(p,d)M_2^2R^2}{BN\varepsilon^2}\right),$$

number of local calls of the gradient-free oracle, $B = 1$ is the number of machines running in parallel and

$$T \sim \frac{1}{\varepsilon^2}: \quad T = NKB = O\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} O\left(\dfrac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2 \quad (q = 2), \\ O\left(\dfrac{M_2^2R^2}{\varepsilon^2}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a two-point gradient-free oracle.

For l_2-randomization, we have the following algorithms:

- **Minibatch Accelerated SGD**

This algorithm, after N communication rounds, gives the convergence rate for $f_\gamma(x)$ (see [11, 22]) in accordance with Corollary 6:

$$\mathbb{E}[f_\gamma(x_{ag}^{N+1}) - f(x_*)] \leq \frac{4L_{f_\gamma}R^2}{N^2} + \frac{4\sigma R}{\sqrt{BKN}} + \frac{\sqrt{d}\Delta R}{\gamma},$$

where $x_*(\gamma) = \underset{x \in Q_\gamma}{\operatorname{argmin}} f_\gamma(x)$.

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(x)$ with $\gamma = \frac{\varepsilon}{2M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(x)$:

$$f(x_{ag}^{N+1}) - f(x_*) \leq f(x_{ag}^{N+1}) - f(x_*(\gamma)) \leq f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) + \gamma M_2 \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(x)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \leq \frac{\varepsilon}{6}, \tag{A.56}$$

$$\frac{4L_{f_\gamma}R^2}{N^2} \leq \frac{\varepsilon}{6}, \tag{A.57}$$

$$\frac{4\sigma R}{\sqrt{BKN}} \leq \frac{\varepsilon}{6}. \tag{A.58}$$

Substituting $L_{f_\gamma} = \frac{\sqrt{d}M}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\varepsilon}{2M_2}$ and $\sigma^2 = 2\kappa(p,d)M_2^2$ (from Corollary 3), into inequalities (A.56)–(A.58), we get

$$\Delta \le \frac{\gamma\varepsilon}{6\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{12\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise,

$$N^2 \ge \frac{48\sqrt{d}MM_2R^2}{\varepsilon^2} \Rightarrow N \ge \frac{4\sqrt{3}d^{1/4}\sqrt{MM_2}R}{\varepsilon} \Rightarrow N = O\left(\frac{d^{1/4}\sqrt{MM_2}R}{\varepsilon}\right),$$

number of communication rounds,

$$B \ge \frac{576\sigma^2R^2}{KN\varepsilon^2} \Rightarrow B \ge \frac{1152\kappa(p,d)M_2^2R^2}{KN\varepsilon^2} \Rightarrow B = O\left(\frac{\kappa(p,d)M_2^2R^2}{KN\varepsilon^2}\right),$$

number of machines running in parallel and

$$T = NKB = \frac{1152\kappa(p,d)M_2^2R^2}{\varepsilon^2}$$

$$\Rightarrow T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} \tilde{O}\left(\dfrac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2 \quad (q = 2), \\ \tilde{O}\left(\dfrac{(\ln d)M_2^2R^2}{\varepsilon^2}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a two-point gradient-free oracle.

• **Single-Machine Accelerated SGD**

This algorithm, after NK iterations, gives the convergence rate for $f_\gamma(x)$ (see [11, 22]) in accordance with Corollary 6:

$$\mathbb{E}[f_\gamma(x_{ag}^{NK+1}) - f(x_*)] \le \frac{4L_{f_\gamma}R^2}{N^2K^2} + \frac{4\sigma R}{\sqrt{NK}} + \frac{\sqrt{d}\Delta R}{\gamma},$$

where $x_*(\gamma) = \underset{x\in Q_\gamma}{\operatorname{argmin}} f_\gamma(x)$.

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(x)$ with $\gamma = \frac{\varepsilon}{2M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(x)$:

$$f(x_{ag}^{N+1}) - f(x_*) \le f(x_{ag}^{N+1}) - f(x_*(\gamma)) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) + \gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(x)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{6}, \tag{A.59}$$

$$\frac{4L_{f_\gamma}R^2}{K^2N^2} \le \frac{\varepsilon}{6}, \tag{A.60}$$

$$\frac{4\sigma R}{\sqrt{KN}} \le \frac{\varepsilon}{6}. \tag{A.61}$$

Substituting $L_{f_\gamma} = \frac{\sqrt{d}M}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\varepsilon}{2M_2}$ and $\sigma^2 = 2\kappa(p,d)M_2^2$ (from Corollary 3), into inequalities (A.59)–(A.61), we get

$$\Delta \le \frac{\gamma\varepsilon}{6\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{12\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise and

$$NK \ge \frac{576\sigma^2R^2}{\varepsilon^2} \Rightarrow NK \ge \frac{1152\kappa(p,d)M_2^2R^2}{\varepsilon^2}.$$

Since N directly depends on K, the number of communication rounds can be taken $N = 1$, then we get:

$$NK = K = O\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right),$$

number of local calls of the gradient-free oracle and

$$T = NKB = \frac{1152\kappa(p,d)M_2^2R^2}{\varepsilon^2}$$

$$\Rightarrow T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} \tilde{O}\left(\frac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2 \quad (q = 2), \\ \tilde{O}\left(\frac{(\ln d)M_2^2R^2}{\varepsilon^2}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a two-point gradient-free oracle.

• **Local-AC-SA**

This algorithm, after N communication rounds, gives the convergence rate for $f_\gamma(x)$ (see [12, 22]) in accordance with Corollary 6:

$$\mathbb{E}[f_\gamma(x_{ag}^{N+1}) - f(x_*)] \le \frac{4L_{f_\gamma}R^2}{K^2N^2} + \frac{4\sigma R}{\sqrt{BKN}} + \frac{\sqrt{d}\Delta R}{\gamma},$$

where $x_*(\gamma) = \underset{x\in Q_\gamma}{\operatorname{argmin}} f_\gamma(x)$.

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(x)$ with $\gamma = \frac{\varepsilon}{2M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(x)$:

$$f(x_{ag}^{N+1}) - f(x_*) \le f(x_{ag}^{N+1}) - f(x_*(\gamma)) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) + \gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(x)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{6}, \tag{A.62}$$

$$\frac{4L_{f_\gamma}R^2}{K^2N^2} \le \frac{\varepsilon}{6}, \tag{A.63}$$

$$\frac{4\sigma R}{\sqrt{BKN}} \le \frac{\varepsilon}{6}. \tag{A.64}$$

Substituting $L_{f_\gamma} = \frac{\sqrt{d}M}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\varepsilon}{2M_2}$ and $\sigma^2 = 2\kappa(p,d)M_2^2$ (from Corollary 3), into inequalities (A.62)–(A.64), we get

$$\Delta \le \frac{\gamma\varepsilon}{6\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{12\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise and

$$N^2K^2 \ge \frac{48\sqrt{d}MM_2R^2}{\varepsilon^2} \Rightarrow NK \ge \frac{4\sqrt{3}d^{1/4}\sqrt{MM_2}R}{\varepsilon}.$$

Since N directly depends on K, the number of communication rounds can be taken $N = 1$, then we get:

$$NK = K = O\left(\frac{d^{1/4}\sqrt{MM_2}R}{\varepsilon}\right)$$

number of local calls of the gradient-free oracle,

$$B \ge \frac{576\sigma^2R^2}{KN\varepsilon^2} \Rightarrow B \ge \frac{1152\kappa(p,d)M_2^2R^2}{KN\varepsilon^2} \Rightarrow B = O\left(\frac{\kappa(p,d)M_2^2R^2}{KN\varepsilon^2}\right),$$

number of machines running in parallel and

$$T = NKB = \frac{1152\kappa(p,d)M_2^2R^2}{\varepsilon^2}$$

$$\Rightarrow T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} \tilde{O}\left(\dfrac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2 \quad (q = 2), \\ \tilde{O}\left(\dfrac{(\ln d)M_2^2R^2}{\varepsilon^2}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a two-point gradient-free oracle.

- **Federated Accelerated SGD (FedAc)**

This algorithm, after N communication rounds, gives the convergence rate for $f_\gamma(x)$ (see [13]) in accordance with Corollary 6:

$$\mathbb{E}[f_\gamma(x_{ag}^{N+1}) - f(x_*)] \le \frac{L_{f_\gamma}R^2}{KN^2} + \frac{\sigma R}{\sqrt{BKN}} + \min\left\{\frac{L_{f_\gamma}^{1/3}\sigma^{2/3}R^{4/3}}{K^{1/3}N}, \frac{L_{f_\gamma}^{1/2}\sigma^{1/2}R^{3/2}}{K^{1/4}N}\right\} + \frac{\sqrt{d}\Delta R}{\gamma},$$

where $x_*(\gamma) = \underset{x\in Q_\gamma}{\operatorname{argmin}} f_\gamma(x)$.

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(x)$ with $\gamma = \frac{\varepsilon}{2M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(x)$:

$$f(x_{ag}^{N+1}) - f(x_*) \le f(x_{ag}^{N+1}) - f(x_*(\gamma)) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) + \gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(x)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{8}, \tag{A.65}$$

$$\frac{L_{f_\gamma}R^2}{KN^2} \le \frac{\varepsilon}{8}, \tag{A.66}$$

$$\frac{\sigma R}{\sqrt{BKN}} \le \frac{\varepsilon}{8}, \tag{A.67}$$

$$\min\left\{\frac{L_{f_\gamma}^{1/3}\sigma^{2/3}R^{4/3}}{K^{1/3}N}, \frac{L_{f_\gamma}^{1/2}\sigma^{1/2}R^{3/2}}{K^{1/4}N}\right\} \le \frac{\varepsilon}{8}. \tag{A.68}$$

Substituting $L_{f_\gamma} = \dfrac{\sqrt{d}M}{\gamma}$ (from Corollary 2), where $\gamma = \dfrac{\varepsilon}{2M_2}$ and $\sigma^2 = 2\kappa(p,d)dM_2^2$ (from Corollary 3), into inequalities (A.65)–(A.68), we get

$$\Delta \le \frac{\gamma\varepsilon}{8\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{16\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise and

$$NK \ge \frac{8^{1/2}K^{1/2}L_{f_\gamma}^{1/2}R}{\varepsilon^{1/2}} \Rightarrow NK \ge \frac{16^{1/2}K^{1/2}d^{1/4}M^{1/2}M_2^{1/2}R}{\varepsilon} \Rightarrow NK \gtrsim \frac{K^{1/2}}{\varepsilon},$$

$$NK \ge \frac{64\sigma^2R^2}{B\varepsilon^2} \Rightarrow NK \ge \frac{72\kappa(p,d)dM_2^2R^2}{B\varepsilon^2} \Rightarrow NK \gtrsim \frac{1}{B\varepsilon^2},$$

and

$$NK \ge \min\left\{\frac{K^{2/3}L_{f_\gamma}^{1/3}\sigma^{2/3}R^{4/3}}{\varepsilon}, \frac{K^{3/4}L_{f_\gamma}^{1/2}\sigma^{1/2}R^{3/2}}{\varepsilon}\right\}$$

$$\Rightarrow NK \ge \min\left\{\frac{2^{1/3}K^{2/3}d^{1/6}(MM_2)^{1/3}\sigma^{2/3}R^{4/3}}{\varepsilon^{4/3}}, \frac{2^{1/2}K^{3/4}d^{1/4}(MM_2)^{1/2}\sigma^{1/2}R^{3/2}}{\varepsilon^{3/2}}\right\}$$

$$\Rightarrow NK \ge \frac{2^{1/3}K^{2/3}d^{1/6}(MM_2)^{1/3}\sigma^{2/3}R^{4/3}}{\varepsilon^{4/3}} \Rightarrow NK \gtrsim \frac{K^{2/3}}{\varepsilon^{4/3}}.$$

Thus, the smallest number of communication rounds, provided that $NK \in [1, \varepsilon^{-2}]$ and $T \in [1, \varepsilon^{-2}]$, will have the form:

$$N \sim \frac{1}{\varepsilon}: \quad N \ge \frac{8L_{f_\gamma}^{1/3}\sigma^{2/3}R^{4/3}}{K^{1/3}\varepsilon} \Rightarrow N = O\left(\frac{d^{1/2}(\kappa(p,d)M)^{1/3}M_2R^{4/3}}{K^{1/3}\varepsilon^{4/3}}\right),$$

then we get:

$$K \sim \frac{1}{\varepsilon}: \quad K \ge \frac{\sigma^2R^2}{BN\varepsilon^2} \Rightarrow K = O\left(\frac{\kappa(p,d)dM_2^2R^2}{BN\varepsilon^2}\right),$$

number of local calls of the gradient-free oracle, $B = 1$—the number of machines running in parallel and

$$T \sim \frac{1}{\varepsilon^2}: \quad T = NKB = \tilde{O}\left(\frac{\kappa(p,d)dM_2^2R^2}{\varepsilon^2}\right) = \begin{cases} \tilde{O}\left(\dfrac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2 \quad (q = 2), \\ \tilde{O}\left(\dfrac{(\ln d)M_2^2R^2}{\varepsilon^2}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a two-point gradient-free oracle.

C. PROOF OF THEOREM 2

In this subsection, we present a complete proof of Theorem 2. To do this, we divide the proof into two parts: the proof for l_1-randomization and the proof for l_2-randomization.

For l_1-randomization, we have the following algorithms:

• **Minibatch SMP**

This algorithm, after N communication rounds, gives a convergence rate for $f_\gamma(x)$ (see Corollary 7) in accordance with Remark 4:

$$\mathbb{E}[f_\gamma(z_N)] \le \max\left\{\frac{7}{4}\frac{LR^2}{N}, 7\frac{\sigma R}{\sqrt{BKN}}\right\} + \frac{d\Delta R}{\gamma}.$$

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(z)$ with $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(z)$:

$$f(z_N) \le f_\gamma(z_N) + \frac{2}{\sqrt{d}}\gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(z)$ you need

$$\frac{d\Delta R}{\gamma} \le \frac{\varepsilon}{4}, \tag{A.69}$$

$$\max\left\{\frac{7}{4}\frac{LR^2}{N}, 7\frac{\sigma R}{\sqrt{BKN}}\right\} \le \frac{\varepsilon}{4}. \tag{A.70}$$

Substituting $L_{f_\gamma} = \frac{dM}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ and $\sigma^2 = 2\kappa(p,d)M_2^2$ (from Corollary 3), into inequalities (A.69) and (A.70), we get

$$\Delta \le \frac{\gamma\varepsilon}{6dR} \Rightarrow \Delta \le \frac{\varepsilon^2}{24\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise and

$$N \ge \frac{784\sigma^2R^2}{BK\varepsilon^2} \Rightarrow N \ge \frac{1568\kappa(p,d)M_2^2R^2}{BK\varepsilon^2}.$$

Since $K = 1$, and N directly depends on B, the number of communication rounds can be taken $N = 1$, then we get:

$$\Rightarrow B = O\left(\frac{\kappa(p,d)M_2^2R^2}{KN\varepsilon^2}\right),$$

number of machines running in parallel and

$$T = NKB = \frac{1568\kappa(p,d)M_2^2R^2}{\varepsilon^2}$$

$$\Rightarrow T = O\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} O\left(\frac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2 \quad (q = 2), \\ O\left(\frac{M_2^2R^2}{\varepsilon^2}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a two-point gradient-free oracle;

• **Single-Machine SMP**

This algorithm, after NK iterations, gives a convergence rate for $f_\gamma(x)$ (see Corollary 7) in accordance with Remark 4:

$$\mathbb{E}[f_\gamma(z_{NK})] \le \max\left\{\frac{7}{4}\frac{LR^2}{KN}, 7\frac{\sigma R}{\sqrt{KN}}\right\} + \frac{\sqrt{d}\Delta R}{\gamma}.$$

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(z)$ with $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(z)$:

$$f(z_N) \le f_\gamma(z_N) + \frac{2}{\sqrt{d}}\gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(z)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{4}, \tag{A.71}$$

$$\max\left\{\frac{7}{4}\frac{LR^2}{KN}, 7\frac{\sigma R}{\sqrt{KN}}\right\} \le \frac{\varepsilon}{4}. \tag{A.72}$$

Substituting $L_{f_\gamma} = \frac{dM}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ and $\sigma^2 = 2\kappa(p,d)M_2^2$ (from Corollary 3), into inequalities (A.71) and (A.72), we get

$$\Delta \le \frac{\gamma\varepsilon}{6dR} \Rightarrow \Delta \le \frac{\varepsilon^2}{24\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise and

$$N \ge \frac{784\sigma^2R^2}{K\varepsilon^2} \Rightarrow N \ge \frac{1568\kappa(p,d)M_2^2R^2}{K\varepsilon^2}.$$

Since N directly depends on K, the number of communication rounds can be taken $N = 1$, then we get:

$$\Rightarrow K = O\left(\frac{\kappa(p,d)M_2^2R^2}{KN\varepsilon^2}\right),$$

number of local calls of the gradient-free oracle and

$$T = NKB = \frac{1568\kappa(p,d)M_2^2R^2}{\varepsilon^2}$$

$$\Rightarrow T = O\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} O\left(\frac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2 \quad (q = 2), \\ O\left(\frac{M_2^2R^2}{\varepsilon^2}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a two-point gradient-free oracle.

For l_2-randomization, we have the following algorithms:

- **Minibatch SMP**

This algorithm, after N communication rounds, gives a convergence rate for $f_\gamma(x)$ (see Corollary 7) in accordance with Remark 4:

$$\mathbb{E}[f_\gamma(z_N)] \le \max\left\{\frac{7}{4}\frac{LR^2}{N}, 7\frac{\sigma R}{\sqrt{BKN}}\right\} + \frac{\sqrt{d}\Delta R}{\gamma}.$$

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(z)$ with $\gamma = \frac{\varepsilon}{2M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(z)$:

$$f(z_N) \le f_\gamma(z_N) + \gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(z)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{4}, \tag{A.73}$$

$$\max\left\{\frac{7}{4}\frac{LR^2}{N}, 7\frac{\sigma R}{\sqrt{BKN}}\right\} \le \frac{\varepsilon}{4}. \tag{A.74}$$

Substituting $L_{f_\gamma} = \frac{\sqrt{d}M}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\varepsilon}{2M_2}$ and $\sigma^2 = 2\kappa(p,d)M_2^2$ (from Corollary 3), into inequalities (A.73) and (A.74), we get

$$\Delta \le \frac{\gamma\varepsilon}{4\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{8\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise and

$$N \ge \frac{784\sigma^2R^2}{BK\varepsilon^2} \Rightarrow N \ge \frac{1568\kappa(p,d)M_2^2R^2}{BK\varepsilon^2}.$$

Since $K = 1$, and N directly depends on B, the number of communication rounds can be taken $N = 1$, then we get:

$$\Rightarrow B = O\left(\frac{\kappa(p,d)M_2^2R^2}{KN\varepsilon^2}\right),$$

number of machines running in parallel and

$$T = NKB = \frac{1568\kappa(p,d)M_2^2R^2}{\varepsilon^2}$$

$$\Rightarrow T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases}\tilde{O}\left(\frac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2 \quad (q = 2),\\ \tilde{O}\left(\frac{(\ln d)M_2^2R^2}{\varepsilon^2}\right), & p = 1 \quad (q = \infty),\end{cases}$$

total number of calls to a two-point gradient-free oracle;

- **Single-Machine SMP**

This algorithm, after NK iterations, gives a convergence rate for $f_\gamma(x)$ (see Corollary 7) in accordance with Remark 4:

$$\mathbb{E}[f_\gamma(z_{NK})] \le \max\left\{\frac{7}{4}\frac{LR^2}{KN}, 7\frac{\sigma R}{\sqrt{KN}}\right\} + \frac{\sqrt{d}\Delta R}{\gamma}.$$

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(z)$ with $\gamma = \frac{\varepsilon}{2M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(z)$:

$$f(z_N) \le f_\gamma(z_N) + \gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(z)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{4}, \tag{A.75}$$

$$\max\left\{\frac{7}{4}\frac{LR^2}{KN}, 7\frac{\sigma R}{\sqrt{KN}}\right\} \le \frac{\varepsilon}{4}. \tag{A.76}$$

Substituting $L_{f_\gamma} = \frac{\sqrt{d}M}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\varepsilon}{2M_2}$ and $\sigma^2 = 2\kappa(p,d)M_2^2$ (from Corollary 3), into inequalities (A.75) and (A.76), we get

$$\Delta \le \frac{\gamma\varepsilon}{4\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{8\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise and

$$N \ge \frac{784\sigma^2R^2}{K\varepsilon^2} \Rightarrow N \ge \frac{1568\kappa(p,d)M_2^2R^2}{K\varepsilon^2}.$$

Since N directly depends on K, the number of communication rounds can be taken $N = 1$, then we get:

$$\Rightarrow K = O\left(\frac{\kappa(p,d)M_2^2R^2}{KN\varepsilon^2}\right),$$

number of local calls of the gradient-free oracle and

$$T = NKB = \frac{1568\kappa(p,d)M_2^2R^2}{\varepsilon^2}$$

$$\Rightarrow T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2R^2}{\varepsilon^2}\right) = \begin{cases} \tilde{O}\left(\frac{dM_2^2R^2}{\varepsilon^2}\right), & p = 2 \quad (q = 2), \\ \tilde{O}\left(\frac{(\ln d)M_2^2R^2}{\varepsilon^2}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a two-point gradient-free oracle.

D. SINGLE-POINT GRADIENT ALGORITHMS

Let's do the same procedure as in Subsection 4.2 to create gradient-free single-point algorithms.

To begin with, using the example of Lemma 10, we show that Lemma 8 also holds for a single-point oracle.

Lemma 10. *For $\nabla f_\gamma(x,\xi,e)$ and $\nabla f_\gamma(x)$ with Assumption 2, the following holds*

$$\mathbb{E}_{\xi,e}[\langle \nabla f_\gamma(x,\xi,e), r\rangle] \ge \langle \nabla f_\gamma(x), r\rangle - \frac{d\Delta E_e[|\langle \operatorname{sgn}(e), r\rangle|]}{\gamma},$$

where ∇f_γ with l_1-randomization;

$$\mathbb{E}_{\xi,e}[\langle \nabla f_\gamma(x,\xi,e), r\rangle] \ge \langle \nabla f_\gamma(x), r\rangle - \frac{d\Delta E_e[|\langle e, r\rangle|]}{\gamma},$$

where ∇f_γ with l_2-randomization.

Proof. Consider

(i) for l_1-randomization (8):

$$\nabla f_\gamma(x,\xi,e) = \frac{d}{\gamma} f_\delta(x+\gamma e,\xi)\operatorname{sgn}(e) = \frac{d}{\gamma}(f(x+\gamma e,\xi) + \delta(x+\gamma e))\operatorname{sgn}(e)$$
$$= \frac{d}{\gamma}(f(x+\gamma e,\xi)\operatorname{sgn}(e) + \delta(x+\gamma e)\operatorname{sgn}(e)).$$

From this equality follows

$$\mathbb{E}_{\xi,e}[\langle \nabla f_\gamma(x,\xi,e), r\rangle] = \frac{d}{\gamma}\mathbb{E}_{\xi,e}[\langle f(x+\gamma e,\xi)\operatorname{sgn}(e), r\rangle] + \frac{d}{\gamma}\mathbb{E}_e[\langle \delta(x+\gamma e)\operatorname{sgn}(e), r\rangle]. \quad (A.77)$$

Applying Lemma 5 to the first term (A.77), we obtain

$$\begin{aligned}\frac{d}{\gamma}\mathbb{E}_{\xi,e}[\langle f(x+\gamma e,\xi)\operatorname{sgn}(e),r\rangle] &= \frac{d}{\gamma}\mathbb{E}_e\left[\langle \mathbb{E}_\xi\left[f(x+\gamma e,\xi)\right]\operatorname{sgn}(e),r\rangle\right]\\ &= \frac{d}{\gamma}\mathbb{E}_e[\langle f(x+\gamma e)\operatorname{sgn}(e),r\rangle] = \langle \nabla f_\gamma(x),r\rangle.\end{aligned} \tag{A.78}$$

For the second term (A.77) with Assumption 2 we get

$$\frac{d}{\gamma}\mathbb{E}_e[\langle \delta(x+\gamma e)\operatorname{sgn}(e),r\rangle] \geq -\frac{d}{\gamma}\Delta\mathbb{E}_e[|\langle \operatorname{sgn}(e),r\rangle|]. \tag{A.79}$$

Using equations (A.78) and (A.79), for equation (A.77), we obtain a statement of the lemma for l_1-randomization.

(ii) for l_2-randomization (9):

$$\begin{aligned}\nabla f_\gamma(x,\xi,e) &= \frac{d}{\gamma}f_\delta(x+\gamma e,\xi)e = \frac{d}{\gamma}(f(x+\gamma e,\xi)+\delta(x+\gamma e))e\\ &= \frac{d}{\gamma}\left(f(x+\gamma e,\xi)e+\delta(x+\gamma e)e\right).\end{aligned}$$

From this equality follows

$$\mathbb{E}_{\xi,e}[\langle \nabla f_\gamma(x,\xi,e),r\rangle] = \frac{d}{\gamma}\mathbb{E}_{\xi,e}[\langle f(x+\gamma e,\xi)e,r\rangle] + \frac{d}{\gamma}\mathbb{E}_e[\langle \delta(x+\gamma e)e,r\rangle]. \tag{A.80}$$

Applying Lemma 6 to the first term (A.80), we obtain

$$\begin{aligned}\frac{d}{\gamma}\mathbb{E}_{\xi,e}[\langle f(x+\gamma e,\xi)e,r\rangle] &= \frac{d}{\gamma}\mathbb{E}_e\left[\langle \mathbb{E}_\xi\left[f(x+\gamma e,\xi)\right]e,r\rangle\right]\\ &= \frac{d}{\gamma}\mathbb{E}_e[\langle f(x+\gamma e)e,r\rangle] = \langle \nabla f_\gamma(x),r\rangle.\end{aligned} \tag{A.81}$$

For the second term (A.80) with Assumption 2 we get

$$\frac{d}{\gamma}\mathbb{E}_e[\langle \delta(x+\gamma e)e,r\rangle] \geq -\frac{d}{\gamma}\Delta\mathbb{E}_e[|\langle e,r\rangle|]. \tag{A.82}$$

Using equations (A.81) and (A.82), for equation (A.80) we obtain a statement of the lemma for l_2-randomization.

Since Lemma 8 holds for a single-point oracle, Corollaries 5–6 and Remark 4 also hold for a single-point oracle. Thus, we can now obtain estimates of the parameters of single-point gradient-free methods for l_1 and l_2-randomization, in the same way as in Subsection 4.2, using Subsection 3.4. In Theorem 3, estimates of gradient-free convex optimization methods are presented, and in Theorem 4, estimates for saddle problems.

Theorem 3. *Smoothing scheme from Section* 3*, applied to problem* (2)*, provides convergence of the following single-point gradient-free algorithms: Minibatch and Single-Machine Accelerated SGD* [11]*, Local-AC-CA* [12] *and FedAc* [13]. *In other words, in order to achieve the accuracy* ε *of solving problem* (2)*, it is necessary to iterate* NK *with the maximum allowable noise level* Δ *and the total number of calls to the gradient-free oracle* T *in accordance with the chosen method and smoothing scheme:*

- *Minibatch Accelerated SGD*

(i) *for* l_1*-randomization* (8):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = O\left(\frac{d^{1/4}\sqrt{MM_2}R}{\varepsilon}\right); \quad K = 1; \quad B = O\left(\frac{\kappa(p,d)M_2^2G^2R^2}{KN\varepsilon^4}\right);$$

$$T = \tilde{O}\left(\frac{\varkappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases}\tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2),\\ \tilde{O}\left(\dfrac{dM_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty),\end{cases}$$

(ii) *for l_2-randomization* (9):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = O\left(\frac{d^{1/4}\sqrt{MM_2}R}{\varepsilon}\right); \quad K = 1; \quad B = O\left(\frac{\varkappa(p,d)M_2^2G^2R^2}{KN\varepsilon^4}\right);$$

$$T = \tilde{O}\left(\frac{\varkappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases}\tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2),\\ \tilde{O}\left(\dfrac{d(\ln d)M_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty),\end{cases}$$

• *Single-Machine Accelerated SGD*

(i) *for l_1-randomization* (8):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = 1; \quad K = O\left(\frac{\varkappa(q,d)d^2G^2R^2}{\varepsilon^4}\right); \quad B = 1;$$

$$T = \tilde{O}\left(\frac{\varkappa(q,d)d^2G^2R^2}{\varepsilon^4}\right) = \begin{cases}\tilde{O}\left(\dfrac{d^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2),\\ \tilde{O}\left(\dfrac{dG^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty),\end{cases}$$

(ii) *for l_2-randomization* (9):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = 1; \quad K = O\left(\frac{\varkappa(q,d)d^2G^2R^2}{\varepsilon^4}\right); \quad B = 1;$$

$$T = \tilde{O}\left(\frac{\varkappa(q,d)d^2G^2R^2}{\varepsilon^4}\right) = \begin{cases}\tilde{O}\left(\dfrac{d^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2),\\ \tilde{O}\left(\dfrac{d(\ln d)G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty),\end{cases}$$

• *Local-AC-SA*

(i) *for l_1-randomization* (8):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = 1; \quad K = O\left(\frac{d^{1/4}\sqrt{MM_2}R}{\varepsilon}\right); \quad B = O\left(\frac{\varkappa(q,d)d^2G^2R^2}{KN\varepsilon^4}\right);$$

$$T = \tilde{O}\left(\frac{\kappa(q,d)d^2G^2R^2}{\varepsilon^4}\right) = \begin{cases}\tilde{O}\left(\frac{d^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2),\\ \tilde{O}\left(\frac{dG^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty),\end{cases}$$

(ii) *for l_2-randomization* (9):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = 1; \quad K = O\left(\frac{d^{1/4}\sqrt{MM_2}R}{\varepsilon}\right); \quad B = O\left(\frac{\kappa(q,d)d^2G^2R^2}{KN\varepsilon^4}\right);$$

$$T = \tilde{O}\left(\frac{\kappa(q,d)d^2G^2R^2}{\varepsilon^4}\right) = \begin{cases}\tilde{O}\left(\frac{d^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2),\\ \tilde{O}\left(\frac{d(\ln d)G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty),\end{cases}$$

• *Federated Accelerated SGD (FedAc)*

(i) *for l_1-randomization* (8):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = O\left(\frac{d^{1/6}(\kappa(q,d)MM_2)^{1/3}G^{2/3}R^{4/3}}{K^{1/3}\varepsilon^2}\right); \quad K = O\left(\frac{\kappa(q,d)d^2G^2R^2}{BN\varepsilon^4}\right); \quad B = 1;$$

$$T = \tilde{O}\left(\frac{\kappa(q,d)d^2G^2R^2}{\varepsilon^4}\right) = \begin{cases}\tilde{O}\left(\frac{d^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2),\\ \tilde{O}\left(\frac{dG^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty),\end{cases}$$

(ii) *for l_2-randomization* (9):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = O\left(\frac{d^{1/6}(\kappa(q,d)MM_2)^{1/3}G^{2/3}R^{4/3}}{K^{1/3}\varepsilon^2}\right); \quad K = O\left(\frac{\kappa(q,d)d^2G^2R^2}{BN\varepsilon^4}\right); \quad B = 1;$$

$$T = \tilde{O}\left(\frac{\kappa(q,d)d^2G^2R^2}{\varepsilon^4}\right) = \begin{cases}\tilde{O}\left(\frac{d^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2),\\ \tilde{O}\left(\frac{d(\ln d)G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty).\end{cases}$$

Proof. Consider the proof for each randomization and each method separately.

For l_1-randomization, we have the following algorithms:

• **Minibatch Accelerated SGD**

This algorithm, after N communication rounds, gives the convergence rate for $f_\gamma(x)$ (see [11, 22]) in accordance with Corollary 5:

$$\mathbb{E}[f_\gamma(x_{ag}^{N+1}) - f(x_*)] \le \frac{4L_{f_\gamma}R^2}{N^2} + \frac{4\sigma R}{\sqrt{BKN}} + \frac{\sqrt{d}\Delta R}{\gamma},$$

where $x_*(\gamma) = \underset{x\in Q_\gamma}{\operatorname{argmin}} f_\gamma(x)$.

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(x)$ with $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(x)$:

$$f_\gamma(x_{ag}^{N+1}) - f(x_*) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) + \frac{2}{\sqrt{d}}\gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(x)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{6}, \tag{A.83}$$

$$\frac{4L_{f_\gamma}R^2}{N^2} \le \frac{\varepsilon}{6}, \tag{A.84}$$

$$\frac{4\sigma R}{\sqrt{BKN}} \le \frac{\varepsilon}{6}. \tag{A.85}$$

Substituting $L_{f_\gamma} = \frac{dM}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ and $\sigma^2 = \frac{4\kappa(p,d)M_2^2G^2}{\varepsilon^2}$ (from Corollary 4), into inequalities (A.83)–(A.85), we get:

$$\Delta \le \frac{\gamma\varepsilon}{6\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{12\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise,

$$N^2 \ge \frac{48\sqrt{d}MM_2R^2}{\varepsilon^2} \Rightarrow N \ge \frac{4\sqrt{3}d^{1/4}\sqrt{MM_2}R}{\varepsilon} \Rightarrow N = O\left(\frac{d^{1/4}\sqrt{MM_2}R}{\varepsilon}\right),$$

number of communication rounds,

$$B \ge \frac{576\sigma^2R^2}{KN\varepsilon^2} \Rightarrow B \ge \frac{2304\kappa(p,d)M_2^2G^2R^2}{KN\varepsilon^4} \Rightarrow B = O\left(\frac{\kappa(p,d)M_2^2G^2R^2}{KN\varepsilon^4}\right),$$

number of machines running in parallel and

$$T = NKB = \frac{2304\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}$$

$$\Rightarrow T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases} \tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2), \\ \tilde{O}\left(\dfrac{dM_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a single-point gradient-free oracle.

- **Single-Machine Accelerated SGD**

This algorithm, after NK iterations, gives the convergence rate for $f_\gamma(x)$ (see [11, 22]) in accordance with Corollary 5:

$$\mathbb{E}[f_\gamma(x_{ag}^{NK+1}) - f(x_*)] \le \frac{4L_{f_\gamma}R^2}{N^2K^2} + \frac{4\sigma R}{\sqrt{NK}} + \frac{\sqrt{d}\Delta R}{\gamma},$$

where $x_*(\gamma) = \underset{x\in Q_\gamma}{\operatorname{argmin}} f_\gamma(x)$.

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(x)$ with $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(x)$:

$$f_\gamma(x_{ag}^{N+1}) - f(x_*) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) + \frac{2}{\sqrt{d}}\gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(x)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{6}, \tag{A.86}$$

$$\frac{4L_{f_\gamma}R^2}{K^2N^2} \le \frac{\varepsilon}{6}, \tag{A.87}$$

$$\frac{4\sigma R}{\sqrt{KN}} \le \frac{\varepsilon}{6}. \tag{A.88}$$

Substituting $L_{f_\gamma} = \frac{dM}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ and $\sigma^2 = \frac{4\kappa(p,d)M_2^2G^2}{\varepsilon^2}$ (from Corollary 4), into inequalities (A.86)–(A.88), we have

$$\Delta \le \frac{\gamma\varepsilon}{6\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{12\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise and

$$NK \ge \frac{576\sigma^2R^2}{\varepsilon^2} \Rightarrow NK \ge \frac{2304\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}.$$

Since N directly depends on K, the number of communication rounds can be taken $N = 1$, then we get:

$$NK = K = O\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right),$$

the number of local calls of the gradient-free oracle and

$$T = NKB = \frac{2304\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}$$

$$\Rightarrow T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases} \tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2), \\ \tilde{O}\left(\dfrac{dM_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a single-point gradient-free oracle.

• **Local-AC-SA**

This algorithm, after N communication rounds, gives the convergence rate for $f_\gamma(x)$ (see [12, 22]) in accordance with Corollary 5:

$$\mathbb{E}[f_\gamma(x_{ag}^{N+1}) - f(x_*)] \le \frac{4L_{f_\gamma}R^2}{K^2N^2} + \frac{4\sigma R}{\sqrt{BKN}} + \frac{\sqrt{d}\Delta R}{\gamma},$$

where $x_*(\gamma) = \underset{x\in Q_\gamma}{\operatorname{argmin}} f_\gamma(x)$.

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(x)$ with $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(x)$:

$$f_\gamma(x_{ag}^{N+1}) - f(x_*) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) + \frac{2}{\sqrt{d}}\gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(x)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{6}, \tag{A.89}$$

$$\frac{4L_{f_\gamma}R^2}{K^2N^2} \le \frac{\varepsilon}{6}, \tag{A.90}$$

$$\frac{4\sigma R}{\sqrt{BKN}} \le \frac{\varepsilon}{6}. \tag{A.91}$$

Substituting $L_{f_\gamma} = \frac{dM}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ and $\sigma^2 = \frac{4\kappa(p,d)M_2^2G^2}{\varepsilon^2}$ (from Corollary 4), into inequalities (A.89)–(A.91), we get

$$\Delta \le \frac{\gamma\varepsilon}{6\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{12\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise and

$$N^2K^2 \ge \frac{48\sqrt{d}MM_2R^2}{\varepsilon^2} \Rightarrow NK \ge \frac{4\sqrt{3}d^{1/4}\sqrt{MM_2}R}{\varepsilon}.$$

Since N directly depends on K, the number of communication rounds can be taken $N = 1$, then we get:

$$NK = K = O\left(\frac{d^{1/4}\sqrt{MM_2}R}{\varepsilon}\right),$$

number of local calls of the gradient-free oracle,

$$B \ge \frac{576\sigma^2R^2}{KN\varepsilon^2} \Rightarrow B \ge \frac{2304\kappa(p,d)M_2^2G^2R^2}{KN\varepsilon^4} \Rightarrow B = O\left(\frac{\kappa(p,d)M_2^2G^2R^2}{KN\varepsilon^4}\right),$$

number of machines running in parallel and

$$T = NKB = \frac{2304\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}$$

$$\Rightarrow T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases} \tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2), \\ \tilde{O}\left(\dfrac{dM_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a single-point gradient-free oracle.

• **Federated Accelerated SGD (FedAc)**

This algorithm, after N communication rounds, gives the convergence rate for $f_\gamma(x)$ (see [13]) in accordance with Corollary 5:

$$\mathbb{E}[f_\gamma(x_{ag}^{N+1}) - f(x_*)] \le \frac{L_{f_\gamma}R^2}{KN^2} + \frac{\sigma R}{\sqrt{BKN}} + \min\left\{\frac{L_{f_\gamma}^{1/3}\sigma^{2/3}R^{4/3}}{K^{1/3}N}, \frac{L_{f_\gamma}^{1/2}\sigma^{1/2}R^{3/2}}{K^{1/4}N}\right\} + \frac{\sqrt{d}\Delta R}{\gamma},$$

where $x_*(\gamma) = \underset{x\in Q_\gamma}{\operatorname{argmin}} f_\gamma(x)$.

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(x)$ with $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(x)$:

$$f_\gamma(x_{ag}^{N+1}) - f(x_*) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) + \frac{2}{\sqrt{d}}\gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(x)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{8}, \tag{A.92}$$

$$\frac{L_{f_\gamma}R^2}{KN^2} \le \frac{\varepsilon}{8}, \tag{A.93}$$

$$\frac{\sigma R}{\sqrt{BKN}} \le \frac{\varepsilon}{8}, \tag{A.94}$$

$$\min\left\{\frac{L_{f_\gamma}^{1/3}\sigma^{2/3}R^{4/3}}{K^{1/3}N}, \frac{L_{f_\gamma}^{1/2}\sigma^{1/2}R^{3/2}}{K^{1/4}N}\right\} \le \frac{\varepsilon}{8}. \tag{A.95}$$

Substituting $L_{f_\gamma} = \frac{dM}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ and $\sigma^2 = \frac{4\kappa(p,d)M_2^2G^2}{\varepsilon^2}$ (from Corollary 4), into inequalities (A.92)–(A.95), we get

$$\Delta \le \frac{\gamma\varepsilon}{8\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{16\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise,

$$NK \ge \frac{8^{1/2}K^{1/2}L_{f_\gamma}^{1/2}R}{\varepsilon^{1/2}} \Rightarrow NK \ge \frac{16^{1/2}K^{1/2}d^{1/4}M^{1/2}M_2^{1/2}R}{\varepsilon} \Rightarrow NK \gtrsim \frac{K^{1/2}}{\varepsilon},$$

$$NK \ge \frac{64\sigma^2R^2}{B\varepsilon^2} \Rightarrow NK \ge \frac{256\kappa(p,d)M_2^2G^2R^2}{B\varepsilon^4} \Rightarrow NK \gtrsim \frac{1}{B\varepsilon^4},$$

and

$$NK \ge \min\left\{\frac{K^{2/3}L_{f_\gamma}^{1/3}\sigma^{2/3}R^{4/3}}{\varepsilon}, \frac{K^{3/4}L_{f_\gamma}^{1/2}\sigma^{1/2}R^{3/2}}{\varepsilon}\right\}$$

$$\Rightarrow NK \ge \min\left\{\frac{16K^{2/3}d^{1/6}(MM_2)^{1/3}\kappa(p,d)^{1/3}G^{2/3}R^{4/3}}{\varepsilon^2}, \frac{16K^{3/4}d^{1/4}(MM_2)^{1/2}\kappa(p,d)^{1/4}G^{1/2}R^{3/2}}{\varepsilon^2}\right\}$$

$$\Rightarrow NK \ge \frac{16K^{2/3}d^{1/6}(MM_2)^{1/3}\kappa(p,d)^{1/3}G^{2/3}R^{4/3}}{\varepsilon^2} \Rightarrow NK \gtrsim \frac{K^{2/3}}{\varepsilon^2}.$$

Thus, the smallest number of communication rounds, provided that $NK \in [1, \varepsilon^{-4}]$ and $T \in [1, \varepsilon^{-4}]$, will have the form:

$$N \sim \frac{1}{\varepsilon}: \quad N \ge \frac{8L_{f_\gamma}^{1/3}\sigma^{2/3}R^{4/3}}{K^{1/3}\varepsilon} \Rightarrow N = O\left(\frac{d^{1/6}(\kappa(p,d)MM_2)^{1/3}G^{2/3}R^{4/3}}{K^{1/3}\varepsilon^2}\right),$$

then we get:

$$K \sim \frac{1}{\varepsilon^3}: \quad K \ge \frac{\sigma^2R^2}{BN\varepsilon^2} \Rightarrow K = O\left(\frac{\kappa(p,d)M_2^2G^2R^2}{BN\varepsilon^4}\right),$$

number of local calls of the gradient-free oracle, $B = 1$—the number of machines running in parallel and

$$T \sim \frac{1}{\varepsilon^4}: \quad T = NKB = \tilde{O}\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases} \tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2), \\ \tilde{O}\left(\dfrac{dM_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a single-point gradient-free oracle.

For l_2-randomization, we have the following algorithms:

• **Minibatch Accelerated SGD**

This algorithm, after N communication rounds, gives the convergence rate for $f_\gamma(x)$ (see [11, 22]) in accordance with Corollary 6:

$$\mathbb{E}[f_\gamma(x_{ag}^{N+1}) - f(x_*)] \le \frac{4L_{f_\gamma}R^2}{N^2} + \frac{4\sigma R}{\sqrt{BKN}} + \frac{\sqrt{d}\Delta R}{\gamma},$$

where $x_*(\gamma) = \underset{x \in Q_\gamma}{\operatorname{argmin}} f_\gamma(x)$.

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(x)$ with $\gamma = \frac{\varepsilon}{2M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(x)$:

$$f_\gamma(x_{ag}^{N+1}) - f(x_*) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) + \gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(x)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{6}, \tag{A.96}$$

$$\frac{4L_{f_\gamma}R^2}{N^2} \le \frac{\varepsilon}{6}, \tag{A.97}$$

$$\frac{4\sigma R}{\sqrt{BKN}} \le \frac{\varepsilon}{6}. \tag{A.98}$$

Substituting $L_{f_\gamma} = \frac{\sqrt{d}M}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\varepsilon}{2M_2}$ and $\sigma^2 = \frac{4\kappa(p,d)M_2^2G^2}{\varepsilon^2}$ (from Corollary 4), into inequalities (A.96)–(A.98), we get

$$\Delta \le \frac{\gamma\varepsilon}{6\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{12\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise,

$$N^2 \ge \frac{48\sqrt{d}MM_2R^2}{\varepsilon^2} \Rightarrow N \ge \frac{4\sqrt{3}d^{1/4}\sqrt{MM_2}R}{\varepsilon} \Rightarrow N = O\left(\frac{d^{1/4}\sqrt{MM_2}R}{\varepsilon}\right),$$

number of communication rounds,

$$B \ge \frac{576\sigma^2R^2}{KN\varepsilon^2} \Rightarrow B \ge \frac{2304\kappa(p,d)M_2^2G^2R^2}{KN\varepsilon^4} \Rightarrow B = O\left(\frac{\kappa(p,d)M_2^2G^2R^2}{KN\varepsilon^4}\right),$$

number of machines running in parallel and

$$T = NKB = \frac{2304\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}$$

$$\Rightarrow T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases} \tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2), \\ \tilde{O}\left(\dfrac{d(\ln d)M_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a single-point gradient-free oracle.

• **Single-Machine Accelerated SGD**

This algorithm, after NK iterations, gives the convergence rate for $f_\gamma(x)$ (see [11, 22]) in accordance with Corollary 6:

$$E[f_\gamma(x_{ag}^{NK+1}) - f(x_*)] \le \frac{4L_{f_\gamma}R^2}{N^2K^2} + \frac{4\sigma R}{\sqrt{NK}} + \frac{\sqrt{d}\Delta R}{\gamma},$$

where $x_*(\gamma) = \underset{x\in Q_\gamma}{\operatorname{argmin}} f_\gamma(x)$.

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(x)$ with $\gamma = \frac{\varepsilon}{2M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(x)$:

$$f_\gamma(x_{ag}^{N+1}) - f(x_*) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) + \gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(x)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{6}, \tag{A.99}$$

$$\frac{4L_{f_\gamma}R^2}{K^2N^2} \le \frac{\varepsilon}{6}, \tag{A.100}$$

$$\frac{4\sigma R}{\sqrt{KN}} \le \frac{\varepsilon}{6}. \tag{A.101}$$

Substituting $L_{f_\gamma} = \frac{\sqrt{d}M}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\varepsilon}{2M_2}$ and $\sigma^2 = \frac{4\kappa(p,d)M_2^2G^2}{\varepsilon^2}$ (from Corollary 4), into inequalities (A.99)–(A.101), we get

$$\Delta \le \frac{\gamma\varepsilon}{6\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{12\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise and

$$NK \ge \frac{576\sigma^2R^2}{\varepsilon^2} \Rightarrow NK \ge \frac{2304\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}.$$

Since N directly depends on K, the number of communication rounds can be taken $N = 1$, then we get:

$$NK = K = O\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right),$$

number of local calls of the gradient-free oracle and

$$T = NKB = \frac{2304\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}$$

$$\Rightarrow T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases} \tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2), \\ \tilde{O}\left(\dfrac{d(\ln d)M_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a single-point gradient-free oracle.

• **Local-AC-SA**

This algorithm, after N communication rounds, gives the convergence rate for $f_\gamma(x)$ (see [12, 22]) in accordance with Corollary 6:

$$\mathbb{E}[f_\gamma(x_{ag}^{N+1}) - f(x_*)] \le \frac{4L_{f_\gamma}R^2}{K^2N^2} + \frac{4\sigma R}{\sqrt{BKN}} + \frac{\sqrt{d}\Delta R}{\gamma},$$

where $x_*(\gamma) = \underset{x\in Q_\gamma}{\operatorname{argmin}} f_\gamma(x)$.

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(x)$ with $\gamma = \frac{\varepsilon}{2M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(x)$:

$$f_\gamma(x_{ag}^{N+1}) - f(x_*) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) + \gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(x)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{6}, \tag{A.102}$$

$$\frac{4L_{f_\gamma}R^2}{K^2N^2} \le \frac{\varepsilon}{6}, \tag{A.103}$$

$$\frac{4\sigma R}{\sqrt{BKN}} \le \frac{\varepsilon}{6}. \tag{A.104}$$

Substituting $L_{f_\gamma} = \frac{\sqrt{d}M}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\varepsilon}{2M_2}$ and $\sigma^2 = \frac{4\kappa(p,d)M_2^2G^2}{\varepsilon^2}$ (from Corollary 4), into inequalities (A.102)–(A.104), we get:

$$\Delta \le \frac{\gamma\varepsilon}{6\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{12\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise and

$$N^2K^2 \ge \frac{48\sqrt{d}MM_2R^2}{\varepsilon^2} \Rightarrow NK \ge \frac{4\sqrt{3}d^{1/4}\sqrt{MM_2}R}{\varepsilon}.$$

Since N directly depends on K, the number of communication rounds can be taken $N = 1$, then we get:

$$NK = K = O\left(\frac{d^{1/4}\sqrt{MM_2}R}{\varepsilon}\right),$$

number of local calls of the gradient-free oracle,

$$B \ge \frac{576\sigma^2R^2}{KN\varepsilon^2} \Rightarrow B \ge \frac{2304\kappa(p,d)M_2^2G^2R^2}{KN\varepsilon^4} \Rightarrow B = O\left(\frac{\kappa(p,d)M_2^2G^2R^2}{KN\varepsilon^4}\right),$$

number of machines running in parallel and

$$T = NKB = \frac{2304\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}$$

$$\Rightarrow T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases} \tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2), \\ \tilde{O}\left(\dfrac{d(\ln d)M_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a single-point gradient-free oracle.

• **Federated Accelerated SGD (FedAc)**

This algorithm, after N communication rounds, gives the convergence rate for $f_\gamma(x)$ (see [13]) in accordance with Corollary 6:

$$\mathbb{E}[f_\gamma(x_{ag}^{N+1}) - f(x_*)] \le \frac{L_{f_\gamma}R^2}{KN^2} + \frac{\sigma R}{\sqrt{BKN}} + \min\left\{\frac{L_{f_\gamma}^{1/3}\sigma^{2/3}R^{4/3}}{K^{1/3}N}, \frac{L_{f_\gamma}^{1/2}\sigma^{1/2}R^{3/2}}{K^{1/4}N}\right\} + \frac{\sqrt{d}\Delta R}{\gamma},$$

where $x_*(\gamma) = \underset{x\in Q_\gamma}{\operatorname{argmin}} f_\gamma(x)$.

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(x)$ with $\gamma = \frac{\varepsilon}{2M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(x)$:

$$f_\gamma(x_{ag}^{N+1}) - f(x_*) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) \le f_\gamma(x_{ag}^{N+1}) - f(x_*(\gamma)) + \gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(x)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{8}, \tag{A.105}$$

$$\frac{L_{f_\gamma}R^2}{KN^2} \le \frac{\varepsilon}{8}, \tag{A.106}$$

$$\frac{\sigma R}{\sqrt{BKN}} \le \frac{\varepsilon}{8}, \tag{A.107}$$

$$\min\left\{\frac{L_{f_\gamma}^{1/3}\sigma^{2/3}R^{4/3}}{K^{1/3}N}, \frac{L_{f_\gamma}^{1/2}\sigma^{1/2}R^{3/2}}{K^{1/4}N}\right\} \le \frac{\varepsilon}{8}. \tag{A.108}$$

Substituting $L_{f_\gamma} = \frac{\sqrt{d}M}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\varepsilon}{2M_2}$ and $\sigma^2 = \frac{4\kappa(p,d)M_2^2G^2}{\varepsilon^2}$ (from Corollary 4), into inequalities (A.105)–(A.108), we get

$$\Delta \le \frac{\gamma\varepsilon}{8\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{16\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise,

$$NK \ge \frac{8^{1/2}K^{1/2}L_{f_\gamma}^{1/2}R}{\varepsilon^{1/2}} \Rightarrow NK \ge \frac{16^{1/2}K^{1/2}d^{1/4}M^{1/2}M_2^{1/2}R}{\varepsilon} \Rightarrow NK \gtrsim \frac{K^{1/2}}{\varepsilon},$$

$$NK \ge \frac{64\sigma^2R^2}{B\varepsilon^2} \Rightarrow NK \ge \frac{256\kappa(p,d)M_2^2G^2R^2}{B\varepsilon^4} \Rightarrow NK \gtrsim \frac{1}{B\varepsilon^4}$$

and

$$NK \geq \min\left\{\frac{K^{2/3}L_{f_\gamma}^{1/3}\sigma^{2/3}R^{4/3}}{\varepsilon}, \frac{K^{3/4}L_{f_\gamma}^{1/2}\sigma^{1/2}R^{3/2}}{\varepsilon}\right\}$$

$$\Rightarrow NK \geq \min\left\{\frac{16K^{2/3}d^{1/6}(MM_2)^{1/3}\varkappa(p,d)^{1/3}G^{2/3}R^{4/3}}{\varepsilon^2}, \frac{16K^{3/4}d^{1/4}(MM_2)^{1/2}\varkappa(p,d)^{1/4}G^{1/2}R^{3/2}}{\varepsilon^2}\right\}$$

$$\Rightarrow NK \geq \frac{16K^{2/3}d^{1/6}(MM_2)^{1/3}\varkappa(p,d)^{1/3}G^{2/3}R^{4/3}}{\varepsilon^2} \Rightarrow NK \gtrsim \frac{K^{2/3}}{\varepsilon^2}.$$

Thus, the smallest number of communication rounds, provided that $NK \in [1, \varepsilon^{-4}]$ and $T \in [1, \varepsilon^{-4}]$, will have the form:

$$N \sim \frac{1}{\varepsilon}: \quad N \geq \frac{8L_{f_\gamma}^{1/3}\sigma^{2/3}R^{4/3}}{K^{1/3}\varepsilon} \Rightarrow N = O\left(\frac{d^{1/6}(\varkappa(p,d)MM_2)^{1/3}G^{2/3}R^{4/3}}{K^{1/3}\varepsilon^2}\right),$$

then we get:

$$K \sim \frac{1}{\varepsilon^3}: \quad K \geq \frac{\sigma^2R^2}{BN\varepsilon^2} \Rightarrow K = O\left(\frac{\varkappa(p,d)M_2^2G^2R^2}{BN\varepsilon^4}\right)$$

number of local calls of the gradient-free oracle, $B = 1$—the number of machines running in parallel and

$$T \sim \frac{1}{\varepsilon^4}: \quad T = NKB = \tilde{O}\left(\frac{\varkappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases}\tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2),\\ \tilde{O}\left(\dfrac{d(\ln d)M_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty),\end{cases}$$

total number of calls to a single-point gradient-free oracle.

Theorem 4. *The smoothing scheme from Section* 3*, applied to the saddle problem* (*see Remark* 1)*, ensures the convergence of the following single-point gradient-free algorithms: Minibatch SMP and Single-Machine SMP from Appendix* A*. In other words, to achieve the accuracy* ε *of solving the saddle problem* (*see Remark* 1)*, it is necessary to iterate* NK *with the maximum allowable noise level* Δ *and the total number of calls to the gradient-free oracle* T *in accordance with the selected smoothing method and scheme:*

- *Minibatch SMP*

(i) *for* l_1*-randomization* (8):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = 1; \quad K = 1; \quad B = O\left(\frac{\varkappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right);$$

$$T = \tilde{O}\left(\frac{\varkappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases}\tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2),\\ \tilde{O}\left(\dfrac{dM_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty),\end{cases}$$

(ii) *for* l_2*-randomization* (9):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = 1; \quad K = 1; \quad B = O\left(\frac{\varkappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right);$$

$$T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases}\tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2),\\ \tilde{O}\left(\dfrac{d(\ln d)M_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty),\end{cases}$$

• *Single-Machine SMP*

(i) *for l_1-randomization* (8):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = 1; \quad K = O\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right); \quad B = 1;$$

$$T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases}\tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2),\\ \tilde{O}\left(\dfrac{dM_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty),\end{cases}$$

(ii) *for l_2-randomization* (9):

$$\Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right);$$

$$N = 1; \quad K = O\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right); \quad B = 1;$$

$$T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases}\tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2),\\ \tilde{O}\left(\dfrac{d(\ln d)M_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty).\end{cases}$$

Proof. Consider the proof for each randomization and each method separately.

For l_1-randomization, we have the following algorithms:

• **Minibatch SMP**

This algorithm, after N communication rounds, gives a convergence rate for $f_\gamma(x)$ (see Corollary 7) in accordance with Remark 4:

$$E[f_\gamma(z_N)] \le \max\left\{\frac{7}{4}\frac{LR^2}{N}, 7\frac{\sigma R}{\sqrt{BKN}}\right\} + \frac{\sqrt{d}\Delta R}{\gamma}.$$

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(z)$ with $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(z)$:

$$f(z_N) \le f_\gamma(z_N) + \frac{2}{\sqrt{d}}\gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(z)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{4}, \tag{A.109}$$

$$\max\left\{\frac{7}{4}\frac{LR^2}{N}, 7\frac{\sigma R}{\sqrt{BKN}}\right\} \le \frac{\varepsilon}{4}. \tag{A.110}$$

Substituting $L_{f_\gamma} = \frac{dM}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\varepsilon}{4M_2}$ and $\sigma^2 = \frac{4\kappa(p,d)M_2^2G^2}{\varepsilon^2}$ (from Corollary 4), into inequalities (A.109) and (A.110), we get

$$\Delta \le \frac{\gamma\varepsilon}{4\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{8\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise and

$$N \ge \frac{784\sigma^2R^2}{BK\varepsilon^2} \Rightarrow N \ge \frac{3136\kappa(p,d)M_2^2G^2R^2}{BK\varepsilon^4} \Rightarrow$$

Since $K = 1$, and N directly depends on B, the number of communication rounds can be taken $N = 1$, then we get:

$$\Rightarrow B = O\left(\frac{\kappa(p,d)M_2^2G^2R^2}{KN\varepsilon^4}\right)$$

number of machines running in parallel and

$$T = NKB = \frac{2304\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}$$

$$\Rightarrow T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases} \tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2), \\ \tilde{O}\left(\dfrac{dM_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a single-point gradient-free oracle.

- **Single-Machine SMP**

This algorithm, after NK iterations, gives a convergence rate for $f_\gamma(x)$ (see Corollary 7) in accordance with Remark 4:

$$\mathbb{E}[f_\gamma(z_{NK})] \le \max\left\{\frac{7}{4}\frac{LR^2}{KN}, 7\frac{\sigma R}{\sqrt{KN}}\right\} + \frac{\sqrt{d}\Delta R}{\gamma}.$$

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(z)$ with $\gamma = \frac{\sqrt{d}\varepsilon}{4M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(z)$:

$$f(z_N) \le f_\gamma(z_N) + \frac{2}{\sqrt{d}}\gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(z)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{4}, \tag{A.111}$$

$$\max\left\{\frac{7}{4}\frac{LR^2}{KN}, 7\frac{\sigma R}{\sqrt{KN}}\right\} \le \frac{\varepsilon}{4}. \tag{A.112}$$

Substitute $L_{f_\gamma} = \frac{\sqrt{d}M}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\varepsilon}{4M_2}$ and $\sigma^2 = \frac{4\kappa(p,d)M_2^2G^2}{\varepsilon^2}$ (from Corollary 4), into inequalities (A.111) and (A.112), we get

$$\Delta \le \frac{\gamma\varepsilon}{4\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{8\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise and

$$N \geq \frac{784\sigma^2 R^2}{K\varepsilon^2} \Rightarrow N \geq \frac{3136\kappa(p,d)M_2^2 G^2 R^2}{K\varepsilon^4} \Rightarrow$$

Since N directly depends on K, the number of communication rounds can be taken $N = 1$, then we get:

$$\Rightarrow K = O\left(\frac{\kappa(p,d)M_2^2 G^2 R^2}{KN\varepsilon^4}\right),$$

number of local calls of the gradient-free oracle and

$$T = NKB = \frac{2304\kappa(p,d)M_2^2 G^2 R^2}{\varepsilon^4}$$

$$\Rightarrow T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2 G^2 R^2}{\varepsilon^4}\right) = \begin{cases} \tilde{O}\left(\dfrac{d^2 M_2^2 G^2 R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2), \\ \tilde{O}\left(\dfrac{d M_2^2 G^2 R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a single-point gradient-free oracle.

For l_2-randomization, we have the following algorithms:

• **Minibatch SMP**

This algorithm, after N communication rounds, gives a convergence rate for $f_\gamma(x)$ (see Corollary 7) in accordance with Remark 4:

$$E[f_\gamma(z_N)] \leq \max\left\{\frac{7}{4}\frac{LR^2}{N}, 7\frac{\sigma R}{\sqrt{BKN}}\right\} + \frac{\sqrt{d}\Delta R}{\gamma}.$$

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(z)$ with $\gamma = \frac{\varepsilon}{2M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(z)$:

$$f(z_N) \leq f_\gamma(z_N) + \gamma M_2 \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(z)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \leq \frac{\varepsilon}{4}, \tag{A.113}$$

$$\max\left\{\frac{7}{4}\frac{LR^2}{N}, 7\frac{\sigma R}{\sqrt{BKN}}\right\} \leq \frac{\varepsilon}{4}. \tag{A.114}$$

Substitute $L_{f_\gamma} = \frac{\sqrt{d}M}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\varepsilon}{2M_2}$ and $\sigma^2 = \frac{4\kappa(p,d)M_2^2 G^2}{\varepsilon^2}$ (from Corollary 4), into inequalities (A.113) and (A.114), we get

$$\Delta \leq \frac{\gamma\varepsilon}{4\sqrt{d}R} \Rightarrow \Delta \leq \frac{\varepsilon^2}{8\sqrt{d}M_2 R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2 R}\right)$$

level of noise and

$$N \geq \frac{784\sigma^2 R^2}{BK\varepsilon^2} \Rightarrow N \geq \frac{3136\kappa(p,d)M_2^2 G^2 R^2}{BK\varepsilon^4} \Rightarrow$$

Since $K = 1$, and N directly depends on B, the number of communication rounds can be taken $N = 1$, then

$$\Rightarrow B = O\left(\frac{\kappa(p,d)M_2^2G^2R^2}{KN\varepsilon^4}\right),$$

the number of machines running in parallel and

$$T = NKB = \frac{2304\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}$$

$$\Rightarrow T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases}\tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2),\\ \tilde{O}\left(\dfrac{d(\ln d)M_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty),\end{cases}$$

total number of calls to a single-point gradient-free oracle.

• **Single-Machine SMP**

This algorithm, after NK iterations, gives a convergence rate for $f_\gamma(x)$ (see Corollary 7) in accordance with Remark 4:

$$\mathbb{E}[f_\gamma(z_{NK})] \le \max\left\{\frac{7}{4}\frac{LR^2}{KN}, 7\frac{\sigma R}{\sqrt{KN}}\right\} + \frac{\sqrt{d}\Delta R}{\gamma}.$$

If we have $\frac{\varepsilon}{2}$-accuracy for the function $f_\gamma(z)$ with $\gamma = \frac{\varepsilon}{2M_2}$ (from Corollary 1), then we have ε-accuracy for the function $f(z)$:

$$f(z_N) \le f_\gamma(z_N) + \gamma M_2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

So, in order to have $\frac{\varepsilon}{2}$-accuracy for $f_\gamma(z)$ you need

$$\frac{\sqrt{d}\Delta R}{\gamma} \le \frac{\varepsilon}{4}, \tag{A.115}$$

$$\max\left\{\frac{7}{4}\frac{LR^2}{KN}, 7\frac{\sigma R}{\sqrt{KN}}\right\} \le \frac{\varepsilon}{4}. \tag{A.116}$$

Substitute $L_{f_\gamma} = \frac{\sqrt{d}M}{\gamma}$ (from Corollary 2), where $\gamma = \frac{\varepsilon}{2M_2}$ and $\sigma^2 = \frac{4\kappa(p,d)M_2^2G^2}{\varepsilon^2}$ (from Corollary 4), into inequalities (A.115) and (A.116), we get

$$\Delta \le \frac{\gamma\varepsilon}{4\sqrt{d}R} \Rightarrow \Delta \le \frac{\varepsilon^2}{8\sqrt{d}M_2R} \Rightarrow \Delta = O\left(\frac{\varepsilon^2}{\sqrt{d}M_2R}\right)$$

level of noise and

$$N \ge \frac{784\sigma^2R^2}{K\varepsilon^2} \Rightarrow N \ge \frac{3136\kappa(p,d)M_2^2G^2R^2}{K\varepsilon^4} \Rightarrow$$

Since N directly depends on K, the number of communication rounds can be taken $N = 1$, then we get:

$$\Rightarrow K = O\left(\frac{\kappa(p,d)M_2^2G^2R^2}{KN\varepsilon^4}\right),$$

number of local calls of the gradient-free oracle and

$$T = NKB = \frac{2304\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}$$

$$\Rightarrow T = \tilde{O}\left(\frac{\kappa(p,d)M_2^2G^2R^2}{\varepsilon^4}\right) = \begin{cases} \tilde{O}\left(\dfrac{d^2M_2^2G^2R^2}{\varepsilon^4}\right), & p = 2 \quad (q = 2), \\ \tilde{O}\left(\dfrac{d(\ln d)M_2^2G^2R^2}{\varepsilon^4}\right), & p = 1 \quad (q = \infty), \end{cases}$$

total number of calls to a single-point gradient-free oracle.

FUNDING

The research was supported by the Russian Science Foundation (project no. 23-11-00229), https://rscf.ru/en/project/23-11-00229/.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. A. Gasnikov, A. Novitskii, V. Novitskii, F. Abdukhakimov, D. Kamzolov, A. Beznosikov, M. Takáč, P. Dvurechensky, and B. Gu, "The power of first-order smooth optimization for black-box non-smooth problems," *Proceedings of the 39th International Conference on Machine Learning* (2022). arXiv preprint arXiv:2201.12289
2. A. S. Nemirovski and D. B. Yudin, *Complexity of Problems and Efficiency of Optimization Methods* (Nauka, Moscow, 1979) [in Russian].
3. O. Shamir, "An optimal algorithm for bandit and zero-order convex optimization with two-point feedback," J. Mach. Learn. Res. **18** (1), 1703–1713 (2017).
4. A. Akhavan et al., "A gradient estimator via L1-randomization for online zero-order optimization with two point feedback," (2022). arXiv preprint arXiv:2205.13910
5. A. V. Gasnikov et al., "Gradient-free proximal methods with inexact oracle for convex stochastic nonsmooth optimization problems on the simplex," Autom. Remote Control **77** (11), 2018–2034 (2016).
6. P. Kairouz et al., "Advances and open problems in federated learning," Found. Trends Mach. Learn. **14** (1/2), 1–210 (2021).
7. J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, "Optimal rates for zero-order convex optimization: The power of two function evaluations," IEEE Trans. Inf. Theory **61** (5), 2788–2806 (2015).
8. K. Scheinberg, "Finite difference gradient approximation: To randomize or not?" INFORMS J. Comput. **34** (5), 2384–2388 (2022).
9. A. Beznosikov, E. Gorbunov, and A. Gasnikov, "Derivative-free method for composite optimization with applications to decentralized distributed optimization," IFAC-PapersOnLine **53** (2), 4038–4043 (2020).
10. M. Ledoux, *The Concentration of Measure Phenomenon* (Am. Math. Soc., Providence, R.I., 2001).
11. B. E. Woodworth et al., "The min-max complexity of distributed stochastic convex optimization with intermittent communication," *Conference on Learning Theory* (PMLR, 2021), pp. 4386–4437.
12. B. Woodworth et al., "Is local SGD better than minibatch SGD?" *International Conference on Machine Learning* (PMLR, 2020), pp. 10334–10343.
13. H. Yuan and T. Ma, "Federated accelerated stochastic gradient descent," Adv. Neural Inf. Process. Syst. **33**, 5332–5344 (2020).
14. E. Gorbunov, D. Dvinskikh, and A. Gasnikov, "Optimal decentralized distributed algorithms for stochastic convex optimization" (2019). arXiv preprint arXiv:1911.07363
15. J. C. Duchi, P. L. Bartlett, and M. J. Wainwright, "Randomized smoothing for stochastic optimization," SIAM J. Optim. **22** (2), 674–701 (2012).

16. F. Yousefian, A. Nedić, and U. V. Shanbhag, "On stochastic gradient and subgradient methods with adaptive steplength sequences," Automatica **48** (1), 56–67 (2012).
17. A. V. Gasnikov, E. A. Krymova, A. A. Lagunovskaya, I. N. Usmanova, and F. A. Fedorenko, "Stochastic online optimization. Single-point and multi-point non-linear multi-armed bandits. Convex and strongly-convex case," Autom. Remote Control **78** (2), 224–234 (2017).
18. D. Dvinskikh et al., "Gradient-free optimization for non-smooth minimax problems with maximum value of adversarial noise" (2022). arXiv preprint arXiv:2202.06114
19. A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," *Proceedings of the 16th Annual ACMSIAM Symposium on Discrete Algorithms* (2005), pp. 385–394.
20. P. Dvurechensky, E. Gorbunov, and A. Gasnikov, "An accelerated directional derivative method for smooth stochastic convex optimization," Eur. J. Oper. Res. **290** (2), 601–621 (2021).
21. A. Juditsky, A. Nemirovski, and C. Tauvel, "Solving variational inequalities with stochastic mirrorprox algorithm," Stochastic Syst. **1** (1), 17–58 (2011).
22. G. Lan, "An optimal method for stochastic composite optimization," Math. Program. **133** (1), 365–397 (2012).