ZEROTH-ORDER METHODS FOR NON-SMOOTH STOCHASTIC PROBLEMS UNDER HEAVY-TAILED NOISE

Nail Bashirov
MIPT, IITP RAS
bashirov.nail.work@ya.ru

Alexander Gasnikov
MIPT, Innopolis, ISP RAS
gasnikov@yandex.ru

Aleksandr Lobanov
MIPT, Skoltech, ISP RAS
lobanov.av@mipt.ru

ABSTRACT

Recently gradient-free optimisation methods have become a major tool in reinforcement learning and memory-efficient LLM fine-tuning. Under the standard setting of uniformly bounded noise variance an optimal accelerated algorithm has been derived. However, the assumption of bounded variance is strict and usually is not fulfilled in practice. Therefore, we will relax it, allowing the noise distribution to be heavy-tailed and, thus, broadening the class of problems to be solved. We propose gradient-free algorithms with zeroth-order oracle under adversarial noise with unbounded variance, for non-smooth convex and convex-concave optimisation problems. We apply clipping operator to deal with heavy-tailedness and batching to allow efficient computation via parallelization. Our analysis provides asymptotic bounds for such key parameters as iteration complexity, oracle complexity and maximal adversarial noise level.

1 INTRODUCTION

In this paper, we consider two types of non-smooth stochastic optimisation problems. The first one is (strongly)-convex problems

$$\min_{x \in Q} \left\{ f(x) \triangleq \mathbb{E}_{\xi \sim \mathcal{D}} \left[f(x, \xi) \right] \right\}, \tag{1}$$

where $Q \subseteq \mathbb{R}^d$ is convex set, ξ is a random variable from distribution $\mathcal{D}$. Such problems often arise in machine learning, where $f_\xi(x)$ represents the loss function on the data sample ξ [36].

Another one is convex-concave saddle optimisation problems

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \left\{ f(x, y) \triangleq \mathbb{E}_{\xi \sim \mathcal{D}} \left[f(x, y, \xi) \right] \right\}, \tag{2}$$

where $\mathcal{X} \subseteq \mathbb{R}^{d_x}$ and $\mathcal{Y} \subseteq \mathbb{R}^{d_y}$ are convex sets, and ξ satisfies the same condition from equation 1. To simplify the presentation, we introduce an embedded variable $z \triangleq (x, y)^T \in \mathcal{X} \times \mathcal{Y} \triangleq \mathcal{Z}$ and an operator

$$F(z) \triangleq F(x, y) = \begin{pmatrix} \nabla_x f(x, y) \\ -\nabla_y f(x, y) \end{pmatrix}. \tag{3}$$

Saddle point problems are closely related to equilibrium search and game theory, which in turn are applied to economics [34] and variational inequalities [23]. In such problems, different models/players competitively minimize their loss functions, e.g., see adversarial example games [4], hierarchical reinforcement learning [41, 39], and generative adversarial networks [18].

1.1 MOTIVATION

The vast majority of optimisation algorithms are based on the assumption of the first-order oracle availability. This means that an oracle returns the value of the objective function gradient (possibly affected by noise) at the requested point. However, in some cases, the gradient may be inaccessible (e.g., in the case of non-smoothness of the objective function [33, 17]) or computationally inefficient. Therefore, the gradient-free (also called zeroth-order) optimisation is used, where oracle the value

of the function instead of gradient at a given point. Methods based on such an oracle are widely demanded in a variety of areas in deep machine learning: reinforcement learning [10, 16, 40], especially in the multi-armed bandits [15, 5, 2]; distributed learning [1, 42, 29]; and also in the "black-box" [31, 7] settings. Also, these methods are actively applied in scientific purposes. For example, in mathematical modelling problems and computational mathematics [22], in bio-chemical and medical research [30, 21]. Due to various factors, such as discretisation, randomness within the model, or adversity, gradient-free oracle is assumed to be noisy.

Usually to effectively deal with the tails heaviness, the clipping operator is used [18]. It is essentially a projection onto an euclidean ball, which mitigates the affection from noisy oracle's output, and recently is ubiquitously used in first-order probabilistic algorithms [35, 8, 19].

Probabilistic approach. It was shown in [20], convergence in expectation may give inaccurate results in some cases, in particular, in the presence of adversarial noise. As we study noise with heavy-tails, we will stick to probabilistic algorithms whose convergence is evaluated for some threshold $1 - \beta$, $\beta \in (0, 1]$. These algorithms are more sensitive to noisy oracles, and have recently become increasingly popular [8, 19, 11, 35]. However, there are very few gradient-free algorithms with one-point gradient approximation [32], and there are no algorithms that allow the noise variance to be unbounded.

In the standard setting, the noise has light tails (e. g. sub-Gaussian), i. e. for some random variable ξ from distribution $\mathcal{D}$ the condition $\mathbb{E}\left[\exp\left(\|\xi - \mathbb{E}[\xi]\|_2^2/\sigma^2\right)\right] \leq \exp(1)$ is fulfilled. However, when it comes to the zeroth-order oracles this presumption usually can not be applied. Therefore, in this paper we relax burdensome assumption of noise light-tailedness, considering only its α-moment to be bounded, where $\alpha \in (1, 2]$. This allows to enlarge problem's class that we study. From a practical point of view, noise with unbounded variance is actively studied by the machine learning community. For example, it has been shown in [43, 9] that such noise usually arises in data distribution when training LLMs and GANs, and is common to the reward distribution in multi-armed bandits [5].

1.2 Contribution

In this paper, we present algorithms `ZO-clipped-SSTM-OPF` and `R-ZO-clipped-SSTM-OPF` for one-point feedback gradient approximation, complementing the recent work [26]. For saddle problems, a gradient-free algorithm `ZO-clipped-SEG` is proposed and analysed for different gradient approximations. For each algorithm, we obtain estimates for three main criteria: (i) the number of iterations to converge to a given accuracy with some pre-specified probability, (ii) the number of oracle calls, (iii) the maximal admissible noise level. Achieved asymptotic boundaries are optimal in terms of accuracy ε in deterministic case [32]. Table 1 contains the results obtained in the convergence theorems of the proposed methods.

1.3 Paper organisation

This paper is organised as follows. Section 2 introduces definitions, assumptions and techniques. Section 3 contains the pseudocode and convergence analysis for `(R-)ZO-clipped-SSTM-OPF` (Algorithms 1, 2) and `ZO-clipped-SEG` (Algorithm 3). Also, in this section we introduce zeroth-order oracle, corrupted by deterministic adversarial noise, and an evaluation on the maximum allowable noise level. In the section 4 we provide numerical experiments on the synthetic task (least squares problem), affected by heavy-tailed noise from Levi's distribution and demonstrate its superiority compared to non-accelerated/non-clipped methods.

2 Preliminaries

In this section, we introduce the notation, the assumptions used in our analysis and main techniques, which improve algorithm's convergence.

Notation. We use $\langle x, y\rangle := \sum_{i=1}^{d} x_i y_i$ to denote standard inner product of $x, y \in \mathbb{R}^d$, where x_i and y_i are the i-th component of x and y respectively. We denote Euclidean norm in $\mathbb{R}^d$ as

Algorithm	Setup	Assumptions	GA	IC	OC
Alg. 1	Convex (1)	Ass. 2.1 ($\mu = 0$), 2.3, 2.11	OPF	$\left(\frac{d^{1/4}}{\varepsilon}\right)$	$\left(\frac{d}{\varepsilon^2}\right)^{\frac{\alpha}{\alpha-1}}$
Alg. 1, [27]	Convex (1)	Ass. 2.1 ($\mu = 0$), 2.3, 2.11	TPF	$\left(\frac{d^{1/4}}{\varepsilon}\right)$	$\left(\frac{\sqrt{d}}{\varepsilon}\right)^{\frac{\alpha}{\alpha-1}}$
Alg. 2	Convex (1)	Ass. 2.1 ($\mu > 0$), 2.3, 2.11	OPF	$\left(\frac{d^{1/4}}{\sqrt{\mu\varepsilon}}\right)$	$\left(\frac{d}{\sqrt{\mu\varepsilon^3}}\right)^{\frac{\alpha}{\alpha-1}}$
Alg. 2, [27]	Convex (1)	Ass. 2.1 ($\mu > 0$), 2.3, 2.11	TPF	$\left(\frac{d^{1/4}}{\sqrt{\mu\varepsilon}}\right)$	$\left(\sqrt{\frac{d}{\mu\varepsilon}}\right)^{\frac{\alpha}{\alpha-1}}$
Alg. 3	Saddle (2)	Ass. 2.1 ($\mu = 0$), 2.3, 2.11	OPF	$\left(\frac{d^{1/2}}{\varepsilon^2}\right)$	$\left(\frac{d}{\varepsilon^2}\right)^{\frac{\alpha}{\alpha-1}}$
Alg. 3	Saddle (2)	Ass. 2.1 ($\mu = 0$), 2.3, 2.11	TPF	$\left(\frac{d^{1/2}}{\varepsilon^2}\right)$	$\left(\frac{\sqrt{d}}{\varepsilon}\right)^{\frac{\alpha}{\alpha-1}}$

Table 1: Summary of all algorithms, proposed in this paper. The "Setup" and "Assumptions" columns contains the setups and assumptions used about the objective function (defined in section 2) respectively. The "GA" column shows which gradient approximation was used – the one-point (OPF) (see equation 11) or the two-point one (see equation 12). The "IC" and "OC" columns are responsible iterative and oracle complexities, represented as asymptotic bounds, depending on the dimensionality of the problem d and the desired accuracy of ε.

$\|x\|_2 := \sqrt{\langle x, x\rangle}$. We use the notation $B_2^d(r) := \left\{x \in \mathbb{R}^d : \|x\|_2 \leq r\right\}$ to denote Euclidean ball, $S_2^d(r) := \left\{x \in \mathbb{R}^d : \|x\|_2 = r\right\}$ to denote Euclidean sphere. Operator $\mathbb{E}[\,\cdot\,]$ denotes full expectation.

2.1 Assumptions on objective function

Assumption on the target subset. Although we consider an unconstrained optimisation problem, the analysis does not require the assumptions on the function f to be extended to the entire space. It is sufficient to introduce assumptions only on some convex set $Q \subset \mathbb{R}^d$, since it will be shown in Sections Section 3 that the produced by proposed algorithms values stay in some ball around the solution, which allows us to consider sufficiently larger classes of problems.

Assumption 2.1 (Strong convexity)**.** Function $f(x, \xi)\colon Q \times \mathcal{D} \to R$ is μ-strongly convex if there exists a convex subset $Q \subset \mathbb{R}^d$ and a constant $\mu \geq 0$ such that, for all $x_1, x_2 \in Q$ and fixed ξ, the following holds for all values of $\lambda \in [0, 1]$:

$$f(\lambda x_1 + (1-\lambda)x_2, \xi) \leq \lambda f(x_1, \xi) + (1-\lambda) f(x_2, \xi) - \frac{\mu}{2}\lambda(1-\lambda)\left\|x_1 - x_2\right\|_2^2, \tag{4}$$

It also follows from Assumption 2.1 that the function $f(x)$ is μ-strongly convex on the set Q. Before proceeding to the main assumption of our paper, let us first introduce the following definition.

Definition 2.2 (Expansion of objective set)**.** We will call Q_τ a τ-expansion of set Q if for some $\tau > 0$ it is true that

$$Q_\tau = Q \oplus B_2^d(1), \tag{5}$$

where the operation $\oplus$ is a Minkowski sum.

Now, using the Definition of 2.2 we introduce the main assumption on the objective function about its Lipschitzness and α-moment boundedness.

Assumption 2.3 (Lipschitz-continuity and boundedness of α-moment)**.** A function $f(x, \xi)$ is Lipschitz-continuous with constant $M_2(\xi) > 0$ on the extension set Q_τ, if for all $x_1, x_2 \in Q_\tau$ is satisfied

$$|f(x_1, \xi) - f(x_2, \xi)| \leq M_2(\xi)\left\|x_1 - x_2\right\|_2. \tag{6}$$

Furthermore, there exists $\alpha \in (1, 2]$ and $M_2 > 0$ such that $\mathbb{E}_\xi\left[M_2(\xi)^\alpha\right] = M_2^\alpha$.

Note that for $\alpha = 2$ the latter condition becomes equivalent to the condition of uniformly bounded variance. However, when $\alpha < 2$ such an assumption is not fulfilled and heavy-tails appears.

2.2 NON-SMOOTH OPTIMISATION

When solving problems with a non-smooth objective function, the Gaussian (random) smoothing scheme proposed in [14] is used. It allows us take into a consideration a smoothed function $\hat{f}_\tau(x)$ with a number of properties to which gradient analysis is applicable (i.e., with a first-order oracle), as well as to do batch-parallel gradient computations. Such an approximation of the non-smooth function $f(x, \xi)$ has the form

$$\hat{f}_\tau(x) = \mathbb{E}_{\boldsymbol{u},\xi} \left[f(x + \tau \boldsymbol{u}, \xi) \right], \tag{7}$$

where $\boldsymbol{u} \sim \mathcal{U}(B_2^d(1))$ is sampled from a uniform distribution on a unit Euclidean ball. The following lemma shows key properties of smoothing.

Lemma 2.4 ([17], Theorem 2.1)**.** *Let there exist such $Q \subset \mathbb{R}^d$ and $\tau > 0$ such that Assumptions 2.1 and 2.3 are satisfied, then the function $\hat{f}_\tau(x)$ defined in $equation$ 7 has the following properties:*

1. *It is convex and M_2-Lipschitz on the set Q, and it is fulfilled that*

$$\sup_{x \in Q} |\hat{f}_\tau(x) - f(x)| \leq \tau M_2. \tag{8}$$

2. *It is differentiable on the set Q, and its gradient is equal to*

$$\nabla \hat{f}_\tau(x) = \frac{d}{\tau} \mathbb{E}_{\boldsymbol{e}} \left[f(x + \tau \boldsymbol{e}) \boldsymbol{e} \right], \tag{9}$$

   where $\boldsymbol{e} \sim \mathcal{U}(S_2^d(1))$.

3. *It is L-Lipschitz continuous with $L = \sqrt{d} M_2 \tau^{-1}$ on the set Q, i.e., for all $x_1, x_2 \in Q$ it is fulfilled*

$$\left\| \nabla \hat{f}_\tau(x_1) - \nabla \hat{f}_\tau(x_2) \right\|_2 \leq L \left\| x_1 - x_2 \right\|_2. \tag{10}$$

The standard way of constructing a gradient-free algorithm is based on the application of gradient approximations. In these approximations the exact value of the gradient $\boldsymbol{g}(x, \xi)$ is replaced by its approximation $\boldsymbol{g}(x, \xi, \boldsymbol{e})$ using schemes from computational math. We will consider two basic estimates of $\boldsymbol{g}(x, \xi, \boldsymbol{e})$ [16, 29, 27]:

Definition 2.5 (Approximation schemes)**.** Consider following approximations of $\boldsymbol{g}(x, \xi)$:

1. One-point feedback (OPF):

$$\boldsymbol{g}(x, \xi, \boldsymbol{e}) = \frac{d}{\tau} f(x + \tau \boldsymbol{e}, \xi) \boldsymbol{e}; \tag{11}$$

2. Two-point feedback (TPF):

$$\boldsymbol{g}(x, \xi, \boldsymbol{e}) = \frac{d}{2\tau} \left(f(x + \tau \boldsymbol{e}, \xi) - f(x - \tau \boldsymbol{e}, \xi) \right) \boldsymbol{e}. \tag{12}$$

Remark 2.6. *The advantage of equation 11 is half the number of zeroth-order oracle calls for each gradient estimate computation. However, it requires an additional constraint in Assumption 2.7.*

Assumption 2.7 (Boundedness of objective function)**.** There exists a subset Q and constant $G > 0$ such that for $x \in Q$

$$\mathbb{E}_\xi \left[f(x, \xi)^\alpha \right] \leq G^\alpha.$$

We now show that the proposed approximation schemes in Definition 2.5 are unbiased estimates of the gradient $\nabla \hat{f}_\tau(x)$ of the smoothed function $\hat{f}_\tau(x)$, and also have bounded α-moment.

Lemma 2.8 (Properties of approximation schemes)**.** *Schemes equation 11, equation 12 have following properties:*

1. *vector $\boldsymbol{g}(x, \xi, \boldsymbol{e})$ is an unbiased estimate of the gradient of the smoothed function $\nabla \hat{f}_\tau(x)$:*

$$\mathbb{E}_{\xi,\boldsymbol{e}}\left[\boldsymbol{g}(x, \xi, \boldsymbol{e})\right] = \nabla \hat{f}_\tau(x); \tag{13}$$

2. *norm of the vector $\boldsymbol{g}(x, \xi, \boldsymbol{e})$ in α-th degree is bounded in expectation:*

$$\mathbb{E}_{\xi,\boldsymbol{e}}\left[\|\boldsymbol{g}(x, \xi, \boldsymbol{e})\|_2^\alpha\right] \leq \sigma^\alpha, \tag{14}$$

Detailed proof can be seen in Appendix A. Next, for computational efficiency, we define the batching technique, which allows us to compute $\boldsymbol{g}(x, \xi, \boldsymbol{e})$ in parallel. Similarly to the Lemma 2.8, the unbiasedness and boundedness of the batched version of the gradient can be shown.

Definition 2.9 (Batching)**.** Let B be the size of the batch, $\{\xi_i\}_{i=1}^B$ and $\{\boldsymbol{e}_i\}_{i=1}^B$ be independent realisations of random variables. Then the batched gradient approximation is a vector

$$\boldsymbol{g}^B\left(x, \{\xi_i\}_{i=1}^B, \{\boldsymbol{e}_i\}_{i=1}^B\right) = \frac{1}{B}\sum_{i=1}^B \boldsymbol{g}(x, \xi_i, \boldsymbol{e}_i). \tag{15}$$

Lemma 2.10 ([26], Lemma 3)**.** *The batched gradient approximation $\boldsymbol{g}^B\left(x, \{\xi_i\}_{i=1}^B, \{\boldsymbol{e}_i\}_{i=1}^B\right)$ is an unbiased estimate of the gradient of the smoothed function $\hat{f}_\tau(x)$:*

$$\nabla \hat{f}_\tau(x) = \mathbb{E}_{\xi,\boldsymbol{e}}\left[\boldsymbol{g}(x, \xi, \boldsymbol{e})\right] = \mathbb{E}_{\{\xi_i\}_{i=1}^B, \{\boldsymbol{e}_i\}_{i=1}^B}\left[\boldsymbol{g}^B\left(x, \{\xi\}_{i=1}^B, \{\boldsymbol{e}_i\}_{i=1}^B\right)\right], \tag{16}$$

and has a finite α-moment:

$$\mathbb{E}\left[\left\|\boldsymbol{g}^B\left(x, \{\xi_i\}_{i=1}^B, \{\boldsymbol{e}_i\}_{i=1}^B\right) - \mathbb{E}\left[\boldsymbol{g}^B\left(x, \{\xi_i\}_{i=1}^B, \{\boldsymbol{e}_i\}_{i=1}^B\right)\right]\right\|_2^\alpha\right] \leq \frac{2\sigma^\alpha}{B^{\alpha-1}}. \tag{17}$$

Moreover, as shown in [35], when $Q \neq \mathbb{R}^d$, the L-smoothness of the smoothed function $\nabla_\tau(x)$ on a larger set is required to fulfil the conditions of the Assumption 2.11.

Assumption 2.11 (Extended smoothness)**.** There exists a set $Q \subseteq \mathbb{R}^d$ and constants $\tau, L > 0$ such that for all $x \in Q$ and $x^* = \arg\min_{x \in Q} \hat{f}_\tau(x)$

$$\left\|\nabla \hat{f}_\tau(x)\right\|_2^2 \leq 2L\left|\hat{f}_\tau(x) - \hat{f}_\tau(x^*)\right|. \tag{18}$$

3 Main results

This section contains main results of our work. We implement gradient-free approach in first-order methods presented in the work [35] under Assumption 2.3 of noise with unbounded noise variance. To deal with it we apply the clipping operator, introduced in the article [18].

$$\operatorname{clip}(\boldsymbol{g}, \lambda) = \begin{cases} \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_2}\min(\|\boldsymbol{g}\|_2, \lambda), & \boldsymbol{g} \neq \boldsymbol{0}, \\ \boldsymbol{0}, & \boldsymbol{g} = \boldsymbol{0}. \end{cases} \tag{19}$$

By clipping, we artificially limit the norm of the gradient vector estimation, which allows us to reduce the affection of stochasticity and noise. The core algorithm for convex and strongly-convex cases is Stochastic Similar Triangles Method [12]. We complement the analysis made in the article [26] and consider a one-point gradient approximation scheme equation 11.

The method produces a sequence $\left\{x^k, y^k, z^k\right\}_{k=1}^K$ with parameters $\alpha_{k+1} = \frac{k+2}{2aL}$, $A_{k+1} = A_k + \alpha_k$. For gradient descent step batched and clipped gradient approximation $\boldsymbol{g}^k$ is used. The initial values are set as $x^0 = y^0 = z^0$ and $A_0 = \alpha_0 = 0$. For pseudocode see Appendix B.

$$\begin{aligned} x^{k+1} &= \frac{A_k y^k + \alpha_{k+1} z^k}{A_{k+1}}, \\ z^{k+1} &= z^k - \alpha_{k+1}\boldsymbol{g}^k, \\ y^{k+1} &= \frac{A_k y^k + \alpha_{k+1} z^{k+1}}{A_{k+1}}, \end{aligned}$$

3.1 CONVEX CASE

For convex functions we use the Assumption 2.1 with $\mu = 0$. In order to simplify the formulation of the convergence theorem, we will present an abridged version, see the full one, together with the proof, in Appendix B.1. Here and below notation $R = \left\|x^0 - x^*\right\|_2^2$ denotes the distance from the starting point to the optimum, $B_R(x^*)$ is used for an Euclidean ball of radius R with center at x^* and $\sim$-notation hides a logarithmic factor in asymptotic bounds.

Theorem 3.1 (Convergence of the `ZO-clipped-SSTM-OPF`)**.** *Let Assumptions 2.3, 2.11 and 2.1 are held with parameter $\mu = 0$ on $B_{3R}(x^*)$. Then for some level $\beta \in (0, 1]$, accuracy $\varepsilon > 0$, batch size B, parameters $A = {}^{4K}/_{\beta} \geq 1$, $a = \Theta\left(A^2, {}^{\sqrt{d}GK^{\frac{\alpha+1}{\alpha}}A^{\frac{\alpha-1}{\alpha}}}/_{M_2RB^{\frac{\alpha-1}{\alpha}}}\right)$, $\lambda_k = \Theta\left({}^{R}/_{\alpha_{k+1}A}\right)$, $L = \sqrt{d}M_2\tau^{-1}$ and $\tau = {}^{\varepsilon}/_{4M_2}$, we guarantee that with probability at least $1 - \beta$ the Algorithm 1 will achieve desired accuracy $f(y^K) - f(x^*) < \varepsilon$ after*

$$K = \tilde{\mathcal{O}}\left(\max\left\{\frac{d^{1/4}M_2R}{\varepsilon}, \frac{1}{B}\left(\frac{dGM_2R}{\varepsilon^2}\right)^{\frac{\alpha}{\alpha-1}}\right\}\right) \text{ iterations} \tag{20}$$

and $K \cdot B$ oracle calls. Moreover, with the same probability $\left\{x^k, y^k, z^k\right\}_{k=0}^K$ remains in $B_{2R}(x^*)$.

Remark 3.2. *The optimal values of the parameter L and τ are due to the properties of the smoothed function $\hat{f}_\tau(x)$ by Lemma 2.4. We also note that in order to solve the direct problem equation 1 with desired accuracy ε, it is necessary to solve the smoothed one with accuracy at least $\varepsilon/2$. This observation will be further used in the proof of the Theorem 3.1 in Appendix B.1.*

Corollary 3.3. *Omitting the logarithmic factor depending on β, we obtain a match with the optimal result in non-smooth deterministic setting (see [6]) for the first term, while the second one is optimal in ε for $\alpha \in (1, 2]$ and zeroth-order oracle (see [32]). We also note that increasing the size of the batches is logical only under the condition that the left term is smaller than the right one, thus a following boundary on batch size B can be obtained:*

$$B \leq \left(\frac{d^{\frac{3\alpha+1}{4}}G^\alpha M_2R}{\varepsilon^{\alpha+1}}\right)^{\frac{1}{\alpha-1}}. \tag{21}$$

3.2 STRONGLY-CONVEX CASE

For strongly-convex objective functions Assumption 2.1 is fulfilled with $\mu > 0$. A standard approach in this case is to use the restart technique. That is, at the t-th restart, the `R-ZO-clipped-SSTM-OPF` algorithm uses the returned value $\tilde{x}^t$, as a starting point, and then performs K_{t+1} iterations of the `ZO-clipped-SSTM-OPF` algorithm.

Theorem 3.4 (Convergence of the `R-ZO-clipped-SSTM-OPF`)**.** *Let Assumptions 2.3, 2.11 and 2.1 are held with parameter $\mu > 0$ on $B_{3R}(x^*)$. Let also $N = \left\lceil \log_2 \frac{\mu R^2}{2\varepsilon} \right\rceil$ be the number of restarts. Then at each restart $t = 1, \ldots, N$ of `R-ZO-clipped-SSTM-OPF` the algorithm `ZO-clipped-SSTM-OPF` is run with batch size B_t (chosen via equation 21) for $K_t = \tilde{\Theta}\left\lceil \max\left\{\sqrt{\frac{L_tR_{t-1}^2}{\varepsilon_t}}, \frac{2}{B_t}\left(\frac{dGM_2R_{t-1}}{\varepsilon_t^2}\right)^{\frac{\alpha}{\alpha-1}}\right\}\right\rceil$ iterations with parameters $A_t = {}^{K_tN}/_{\beta}$, $a_t = \Theta\left(\max\left\{A_t^2, \left(dG_t^{\frac{\alpha+1}{\alpha}}A_t^{\frac{\alpha-1}{\alpha}}\right)/\left(\tau_tL_tR_tB_t^{\frac{\alpha-1}{\alpha}}\right)\right\}\right)$, $\lambda_k^t = \Theta\left({}^{R_t}/_{\alpha_{k+1}^tA_t}\right)$, $L_t = \sqrt{d}M_2\tau^{-1}$, $\tau_t = {}^{\varepsilon_t}/_{4M_2}$, $\varepsilon_t = {}^{\mu R_{t-1}^2}/_{4}$ and $R_{t-1} = {}^{R}/_{2^{(t-1)/2}}$. We guarantee for some level $\beta \in (0, 1]$, accuracy $\varepsilon > 0$, that with probability at least $1 - \beta$ the algorithm `R-ZO-clipped-SSTM` will achieve the desired accuracy $f(x^N) - f(x^*) < \varepsilon$ after the*

$$\sum_{t=1}^N K_tB_t = \tilde{\mathcal{O}}\left(\max\left\{\frac{d^{1/4}M_2R}{\sqrt{\mu\varepsilon}}, \left(\frac{dGM_2}{\sqrt{\mu\varepsilon^3}}\right)^{\frac{\alpha}{\alpha-1}}\right\}\right) \tag{22}$$

oracle calls. Moreover, all values after the t-th restart will remain in $B_{2R_{t-1}}(x^*)$.

The obtained complexity bound is optimal (up to logarithms) high-probability complexity bound under heavy-tailed noise Assumption 2.3 for the smooth strongly convex problems, due to the first

term cannot be improved in view of the deterministic lower bound [32], and the second term is optimal according [43].

3.3 SADDLE POINT PROBLEMS

Now, we switch to the saddle point problems setup equation 2 and introduce gradient-free algorithm `ZO-clipped-SEG`. It will be based on the first-order one, presented in the paper [35] for variational inequalities, utilizing the idea of same-step stochastic extra-gradient method [28]. Since this class of problems is broader, than the researched one, the relation of properties and metrics will be shown. To begin with, we introduce `Gap` metric, defined for variational inequalities (VIPs) and saddle point problems (SPPs), respectively.

Definition 3.5 (Gap metric)**.** Let $\tilde{z} \triangleq (\tilde{x}, \tilde{y})^\top$ is a point from set $\mathcal{Z} \triangleq \mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^{d_x+d_y}$, respectively. Then, we define the gap metric for variational inequalities $\texttt{Gap}^{\text{VIP}}$ for all $z \in \mathcal{Z}$ and operator F as

$$\texttt{Gap}^{\text{VIP}}_{\mathcal{Z}}(\tilde{z}) = \max_{z \in \mathcal{Z}} \langle F(z),\, \tilde{z} - z \rangle \,. \tag{23}$$

Also, for all points $x \in \mathcal{X}$, $y \in \mathcal{Y}$ the gap metric for saddle point problems $\texttt{Gap}^{\text{SPP}}$ is defined as

$$\texttt{Gap}^{\text{SPP}}_{\mathcal{X}\times\mathcal{Y}}(\tilde{x}, \tilde{y}) = \max_{(x,y)^\top \in \mathcal{X}\times\mathcal{Y}} [f(\tilde{x}, x) - f(\tilde{y}, y)]. \tag{24}$$

It can be easily shown, that for all $\tilde{z} \in \mathcal{Z}$ the following inequality holds (see [24])

$$\texttt{Gap}^{\text{SPP}}_{\mathcal{Z}}(\tilde{z}) \leq \texttt{Gap}^{\text{VIP}}_{\mathcal{Z}}(\tilde{z}). \tag{25}$$

As we consider non-smooth optimization, we cannot utilize the gradient of the objective function f when defining operator F via equation 3. Therefore, we use the same gradient approximations for gradients (with $\boldsymbol{e} = (\boldsymbol{e}_x, -\boldsymbol{e}_y)^\top$, where $\boldsymbol{e}_x \sim S_2^d(1), \boldsymbol{e}_y \sim S_2^d(1)$) and apply the same assumptions and techniques introduced in Section 2 to the function $f(z, \xi)$ subject to embedded variable z.

The update rules of the `ZO-clipped-SEG` can be written as follows. Here $\tilde{F}$ stands for clipped and batched operator F, $\xi_1^k, \xi_2^k \sim \mathcal{D}^k$ and $\boldsymbol{e}_1^k, \boldsymbol{e}_2^k \sim S_2^d(1)$ are sampled independently.

$$\boxed{\begin{aligned} \tilde{z}^k &= z^k - \gamma \tilde{F}(z^k, \xi_1^k, \boldsymbol{e}_1^k) \\ z^k &= \tilde{z}^k - \gamma \tilde{F}(\tilde{z}^k, \xi_2^k, \boldsymbol{e}_2^k) \end{aligned}}$$

Theorem 3.6 (Convergence of the `ZO-clipped-SEG`)**.** *Let Assumptions 2.3 with $\mu = 0$, 2.1 and 2.11 be satisfied on the set $B_{4R}(z^*)$. Then for some level $\beta \in (0, 1]$, accuracy ε, parameters $A = \ln {}^{6(K+1)}/_{\beta}$, $\gamma = \Theta\left(\min\left\{(AL)^{-1}, {}^{R}/\left(\sigma K^{\frac{1}{\alpha}} A^{\frac{\alpha-1}{\alpha}}\right)\right\}\right)$, $\lambda_k = \Theta({}^{R}/_{\gamma A})$, $L = \sqrt{d} M_2 \tau^{-1}$, $\tau = {}^{\varepsilon}/_{4M_2}$ and σ defined by the equations equation 34-equation 35, we guarantee that with probability at least $1 - \beta$ the algorithm 3 will achieve desired accuracy $\texttt{Gap}^{\text{SPP}}_{B_R(z^*)}\left(\frac{1}{K}\sum\limits_{k=0}^{K} \tilde{z}_k\right) \leq \varepsilon$ after where*

$$K = \mathcal{O}\left(\max\left\{\frac{\sqrt{d} M_2^2 R^2}{\varepsilon^2} \ln \frac{\sqrt{d} M_2^2 R^2}{\varepsilon^2 \beta}, \left(\frac{\sigma R}{\varepsilon}\right)^{\frac{\alpha}{\alpha-1}} \ln \frac{\sigma R}{\varepsilon \beta}\right\}\right) \text{ iterations.} \tag{26}$$

Pseudocode, together with the full version and proof can be found in the Appendix B.3. Obtained iterative boundary is optimal for deterministic case [3, 37]. However, substituting σ from the Lemma 2.4, one can note a non-optimal oracle complexity. The optimal estimate in the case of unbounded variance for the gradient-free oracle is $d\varepsilon^{-\frac{\alpha}{\alpha-1}}$, which differs from the estimate achieved in [32], being optimal in terms of d for OPF scheme and optimal in terms of ε for TPF one. Nonetheless, as far as we know, the proposed asymptotic boundary is the first one for convex-concave saddle problems.

3.4 ZEROTH-ORDER ORACLE CORRUPTED BY ADVERSARIAL NOISE

In the "black-box" optimization setting, the zeroth-order oracle is usually affected by some deterministic adversarial noise $\delta(x, \xi)$.

Definition 3.7 (Biased zeroth-order oracle)**.** The gradient-free oracle returns the value of the function $f(x, \xi)$ at the requested point x, with some noise $\delta(x, \xi)$:

$$f_\delta(x, \xi) = f(x, \xi) + \delta(x, \xi). \tag{27}$$

This formulation is natural since inaccuracies can arise, for example, due to discretisation of the values of $f(x, \xi)$, or in case of differential privacy. Such oracles have been widely studied with deterministic [29, 11, 38] or stochastic [29, 20] noise functions δ. We will also introduce a common assumption on boundedness of δ.

Assumption 3.8 (Bounded noise)**.** There exists $\Delta > 0$ such that for all $x \in Q$ and fixed $\xi \sim \mathcal{D}$

$$\|\delta(x, \xi)\|_2 \leq \Delta. \tag{28}$$

It is a standard assumption used in setups of adversarial attacks. The larger Δ is, the cheaper it is to invoke the oracle, since the function value can be calculated with less accuracy. However, a noise term affects the gradient approximations. Therefore, to guarantee the convergence rates presented in the Theorems 3.1, 3.4 and 3.6, it is necessary for the gradient approximations to be unbiased and the variance should not depend on the noise level, i.e., the deterministic term should dominate the noise one. From this statement we derive the boundaries for maximal admissible noise level.

Lemma 3.9 (Maximal admissible noise level)**.** *To guarantee the convergence rate presented in the Theorems 3.1, 3.4 and 3.6, the noise level Δ must fulfil the following constraints:*

$$\textit{Noisy OPF:} \qquad \Delta \lesssim \min\left\{G, \frac{\varepsilon^2}{\sqrt{d} M_2 R}\right\}, \tag{29}$$

$$\textit{Noisy TPF:} \qquad \Delta \lesssim \frac{\varepsilon^2}{\sqrt{d} M_2 R}. \tag{30}$$

Remark 3.10 (μ-strongly convex case)**.** *Following the result, obtained in the work [11], for μ-strongly convex case the $\varepsilon^2/\sqrt{d} M_2 R$ is changed to $\sqrt{\mu\varepsilon^3}/\sqrt{d} M_2$.*

4 Experiments

This part contains the results of numerical experiments for the `ZO-clipped-SSTM-OPF` method. We solve the quadratic minimisation problem:

$$\min_{x \in \mathbb{R}^d} \|Ax - b\|_2 + \langle \xi, x \rangle, \tag{31}$$

where ξ is a random vector with jointly independent components sampled from the Levy α-stable distribution with $\alpha = 1.5$ and $A \in \mathbb{R}^{n \times d}, b \in \mathbb{R}^n$ (in the experiment $d = 8, n = 100$). For the problem equation 31, the Assumption 2.1 with $\mu \geq 0$ and the Assumption 2.3 with the Lipschitz constant $M_2(\xi) = \|A\|_2 + \|\xi\|_2$ are fulfilled.

We compare a non-accelerated methods `ZO-SGD-OPF` and its clipped version `ZO-clipped-SGD-OPF` (see [19]) with the accelerated versions `ZO-SSTM-OPF` and `ZO-clipped-SSTM-OPF` with one-point feedback gradient approximation scheme. In order to find the optimal parameters for batch size B, learning rate γ and clipping parameter λ we apply grid search on the following grid: $B \in \{1, 2, 5, 10, 50\}$, $\gamma \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$, $\lambda \in \{10, 1, 0.1, 0.01\}$.

From the graphs in Fig. 1 we can observe, that clipping technique allows methods to converge to lower error floor.

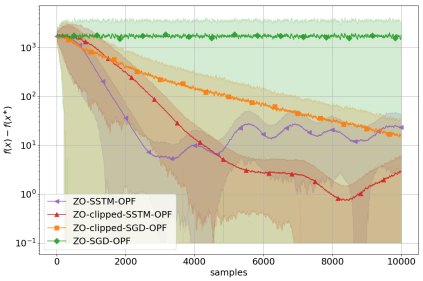

Figure 1: Comparison of accelerated and non-accelerated gradient-free methods with the following parameter choice: `ZO-SGD-OPF`: $B = 10$, $a = 10^{-6}$; `ZO-SSTM-OPF`: $B = 10$, $a = 10^{-6}$; `ZO-clipped-SGD-OPF`: $B = 10$, $a = 10^{-6}$, $\lambda = 0.1$; `ZO-clipped-SSTM-OPF`: $B = 10$, $a = 10^{-6}$, $\lambda = 1.0$.

5 Conclusion

In this article we complement the research conducted in [26], introducing the analysis of one-point feedback gradient approximation scheme for convex and strongly convex setups. Also we present a new gradient-free algorithm `ZO-clipped-SEG` to solve non-smooth convex-concave saddle point

problems. For all algorithms we prove their optimality in terms of desired accuracy ε or problem dimensionality d. Moreover, we provide an analysis of maximal admissible level of noise (maybe adversarial) and numerical experiments, showing the superiority of clipping technique when the noise distribution is heavy tailed.

For future work we aim for providing zeroth-order analysis for composite convex optimization problems and decentralized learning.

REFERENCES

[1] Arya Akhavan, Massimiliano Pontil, and Alexandre B. Tsybakov. Distributed zero-order optimization under adversarial noise, 2021.

[2] Peter Bartlett, Varsha Dani, Thomas Hayes, Sham Kakade, Alexander Rakhlin, and Ambuj Tewari. High-probability regret bounds for bandit online linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory-COLT 2008*, pp. 335–342. Omnipress, 2008.

[3] A. Bayandina, Alexander Gasnikov, and Anastasia Lagunovskaya. Gradient-free two-point methods for solving stochastic nonsmooth convex optimization problems with small non-random noises. *Automation and Remote Control*, 79:1399–1408, 08 2018. doi: 10.1134/S0005117918080039.

[4] A. Bose, Gauthier Gidel, Hugo Berrard, Andre Cianflone, Pascal Vincent, Simon Lacoste-Julien, and William L. Hamilton. Adversarial example games. *ArXiv*, abs/2007.00720, 2020.

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

[6] Sébastien Bubeck, Qijia Jiang, Yin Tat Lee, Yuanzhi Li, and Aaron Sidford. Near-optimal method for highly smooth convex optimization. In Alina Beygelzimer and Daniel Hsu (eds.), *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pp. 492–507. PMLR, 25–28 Jun 2019.

[7] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, CCS '17. ACM, Nov 2017. doi: 10.1145/3128572.3140448.

[8] Ashok Cutkosky and Harsh Mehta. High-probability bounds for non-convex stochastic optimization with heavy tails, 2021.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

[10] Yuriy Dorn, Aleksandr Katrutsa, Ilgam Latypov, and Andrey Pudovikov. Fast ucb-type algorithms for stochastic bandits with heavy and super heavy symmetric noise. *arXiv preprint arXiv:2402.07062*, 2024.

[11] Darina Dvinskikh, Vladislav Tominin, Iaroslav Tominin, and Alexander Gasnikov. Noisy zeroth-order optimization for non-smooth saddle point problems. In *International Conference on Mathematical Optimization Theory and Operations Research*, pp. 18–33. Springer, 2022.

[12] Pavel E. Dvurechensky, Alexander V. Gasnikov, and Alexander Tiurin. Randomized similar triangles method: A unifying framework for accelerated randomized optimization methods (coordinate descent, directional search, derivative-free method). *ArXiv*, abs/1707.08486, 2017.

[13] Pavel E. Dvurechensky, Alexander V. Gasnikov, and Eduard A. Gorbunov. An accelerated directional derivative method for smooth stochastic convex optimization. *Eur. J. Oper. Res.*, 290:601–621, 2018.

[14] Yuri Ermoliev. *Stochastic Programming Methods*. Nauka, 1976.

[15] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *arXiv preprint cs/0408007*, 2004.

[16] Alexander Gasnikov, Darina Dvinskikh, Pavel Dvurechensky, Eduard Gorbunov, Aleksandr Beznosikov, and Alexander Lobanov. *Randomized Gradient-Free Methods in Convex Optimization*, pp. 1–15. Springer International Publishing, Cham, 2020. ISBN 978-3-030-54621-2. doi: 10.1007/978-3-030-54621-2_859-1.

[17] Alexander V. Gasnikov, Anton Novitskii, Vasilii Novitskii, Farshed Abdukhakimov, Dmitry Kamzolov, Aleksandr Beznosikov, Martin Takác, Pavel E. Dvurechensky, and Bin Gu. The power of first-order smooth optimization for black-box non-smooth problems. In *International Conference on Machine Learning*, 2022.

[18] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems*, 2014.

[19] Eduard Gorbunov, Marina Danilova, Innokentiy Shibaev, Pavel Dvurechensky, and Alexander Gasnikov. Near-optimal high probability complexity bounds for non-smooth stochastic optimization with heavy-tailed noise, 2022.

[20] Eduard A. Gorbunov, Marina Danilova, and Alexander V. Gasnikov. Stochastic optimization with heavy-tailed noise via accelerated gradient clipping. *ArXiv*, abs/2005.10785, 2020.

[21] Genetha Gray, Tamara Kolda, Kenneth Sale, and Malin Young. Optimizing an empirical scoring function for transmembrane protein structure determination. *INFORMS Journal on Computing*, 16:406–418, Dec 2004. doi: 10.1287/ijoc.1040.0102.

[22] Jihun Han, Mihai Nica, and Adam R. Stinchcombe. A derivative-free method for solving elliptic partial differential equations with deep neural networks. *Journal of Computational Physics*, 419:109672, Oct 2020. ISSN 0021-9991. doi: 10.1016/j.jcp.2020.109672.

[23] Patrick Harker and Jong-Shi Pang. Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications. *Math. Program.*, 48:161–220, 03 1990. doi: 10.1007/BF01582255.

[24] Anatoli Juditsky, Arkadi Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Siam Journal on Control and Optimization - SIAM*, 1, 09 2008. doi: 10.1214/10-SSY011.

[25] Nikita Kornilov, Alexander V. Gasnikov, Pavel E. Dvurechensky, and Darina Dvinskikh. Gradient-free methods for non-smooth convex stochastic optimization with heavy-tailed noise on convex compact. *Computational Management Science*, 20:1–43, 2023.

[26] Nikita Kornilov, Ohad Shamir, Aleksandr Lobanov, Darina Dvinskikh, Alexander V. Gasnikov, Innokentiy Shibaev, Eduard A. Gorbunov, and Samuel Horv'ath. Accelerated zeroth-order method for non-smooth stochastic convex optimization problem with infinite variance. In *Neural Information Processing Systems*, 2023.

[27] Nikita Kornilov, Yuriy Dorn, Aleksandr Lobanov, Nikolay Kutuzov, Innokentiy Shibaev, Eduard Gorbunov, Alexander Gasnikov, and Alexander Nazin. Zeroth-order median clipping for non-smooth convex optimization problems with heavy-tailed symmetric noise. *arXiv preprint arXiv:2402.02461*, 2024.

[28] G. M. Korpelevich. The extragradient method for finding saddle points and other problems. 1976.

[29] Aleksandr Lobanov, Andrew Veprikov, Georgiy Konin, Aleksandr Beznosikov, Alexander Gasnikov, and Dmitry Kovalev. Non-smooth setting of stochastic decentralized convex optimization problem over time-varying graphs. *Computational Management Science*, 20(1), 2023. ISSN 1619-6988. doi: 10.1007/s10287-023-00479-7.

[30] Alison Marsden, Jeffrey Feinstein, and Charles Taylor. A computational framework for derivative-free optimization of cardiovascular geometries. *Computer Methods in Applied Mechanics and Engineering*, 197:1890–1905, 2008.

[31] N. Narodytska and S. Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1310–1318, Los Alamitos, CA, USA, Jul 2017. IEEE Computer Society. doi: 10.1109/CVPRW.2017.172.

[32] A. S. Nemirovsky and D. B. Yudin. Problem complexity and method efficiency in optimization. *The Journal of the Operational Research Society*, 35(5):455–455, 1984. ISSN 01605682, 14769360.

[33] Boris Teodorovich Polyak. Minimization of unsmooth functionals. *USSR Computational Mathematics and Mathematical Physics*, 9(3):14–29, 1969.

[34] E. Roewland. Theory of games and economic behavior. *Nature*, 157(3981):172–173, Feb 1946. ISSN 1476-4687. doi: 10.1038/157172a0.

[35] Abdurakhmon Sadiev, Marina Danilova, Eduard Gorbunov, Samuel Horváth, Gauthier Gidel, Pavel Dvurechensky, Alexander Gasnikov, and Peter Richtárik. High-probability bounds for stochastic optimization and variational inequalities: the case of unbounded variance, 2023.

[36] Shai Shalev-Shwartz and Shai Ben-David. Understanding machine learning: From theory to algorithms. *Understanding Machine Learning: From Theory to Algorithms*, 01 2013. doi: 10.1017/CBO9781107298019.

[37] Ohad Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *Journal of Machine Learning Research*, 18, 07 2015.

[38] Ekaterina Statkevich, Sofiya Bondar, Darina Dvinskikh, Alexander Gasnikov, and Aleksandr Lobanov. Gradient-free algorithm for saddle point problems under overparametrization. *Chaos, Solitons & Fractals*, 185:115048, August 2024. ISSN 0960-0779.

[39] Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning, 2017.

[40] Jingkang Wang, Yang Liu, and Bo Li. Reinforcement learning with perturbed rewards. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6202–6209, 2020. doi: 10.1609/aaai.v34i04.6086.

[41] Greg Wayne and L.F. Abbott. Hierarchical control using networks trained with higher-level forward models. *Neural computation*, 26:1–31, 07 2014.

[42] Zhan Yu, Daniel W. C. Ho, and Deming Yuan. Distributed randomized gradient-free mirror descent algorithm for constrained optimization, 2019.

[43] Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank J Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models?, 2020.

A TECHNICAL LEMMAS

Lemma A.1 (Lemma 2.8)**.** *Schemes equation 11, equation 12 have following properties:*

1. *vector $\boldsymbol{g}(x, \xi, \boldsymbol{e})$ is an unbiased estimate of the gradient of the smoothed function $\nabla \hat{f}_\tau(x)$:*

$$\mathbb{E}_{\xi,\boldsymbol{e}}\left[\boldsymbol{g}(x, \xi, \boldsymbol{e})\right] = \nabla \hat{f}_\tau(x); \tag{32}$$

2. *norm of the vector $\boldsymbol{g}(x, \xi, \boldsymbol{e})$ in α-th degree is bounded in expectation:*

$$\mathbb{E}_{\xi,\boldsymbol{e}}\left[\|\boldsymbol{g}(x, \xi, \boldsymbol{e})\|_2^\alpha\right] \leq \sigma^\alpha, \tag{33}$$

where parameter σ has following values for

$$\textit{OPF}: \qquad \sigma = \frac{dG}{\tau}, \tag{34}$$

$$\textit{TPF}: \qquad \sigma = \frac{\sqrt{d}M_2}{2^{1/4}}. \tag{35}$$

Proof. The unbiasedness of the gradient estimates can be shown directly by the Definition 2.5:

1.1 For OPF scheme we have

$$\mathbb{E}_{\xi,\boldsymbol{e}}\left[\boldsymbol{g}(x, \xi, \boldsymbol{e})\right] = \mathbb{E}_{\xi,\boldsymbol{e}}\left[\frac{d}{\tau} f(x + \tau\boldsymbol{e}, \xi)\boldsymbol{e}\right] \overset{equation\ 9}{=} \nabla \hat{f}_\tau(x). \tag{36}$$

1.2 For TPF, the reasoning is similar and is based on the symmetry of the distribution of the vector $\boldsymbol{e}$, and independence with ξ.

$$\begin{aligned}\mathbb{E}_{\xi,\boldsymbol{e}}\left[\boldsymbol{g}(x, \xi, \boldsymbol{e})\right] &= \mathbb{E}_{\xi,\boldsymbol{e})}\left[\frac{d}{2\tau}(f(x + \tau\boldsymbol{e}, \xi) - f(x - \tau\boldsymbol{e}, \xi))\boldsymbol{e}\right] \\ &= \frac{d}{2\tau}\mathbb{E}_{\boldsymbol{e}}\left[\mathbb{E}_{\xi}\left[f(x + \tau e, \xi)\boldsymbol{e}\right] + \mathbb{E}_{\xi}\left[f(x - \tau e, \xi)\boldsymbol{e}\right]\right] \\ &= \frac{d}{\tau}\mathbb{E}_{\boldsymbol{e}}\left[f(x + \tau e)\boldsymbol{e}\right] = \nabla \hat{f}_\tau(x).\end{aligned} \tag{37}$$

2.1 To prove the boundedness of the gradient norm for the OPF we use an Assumption 2.7:

$$\begin{aligned}\mathbb{E}_{\xi,\boldsymbol{e}}\left[\|\boldsymbol{g}(x, \xi, \boldsymbol{e})\|_2^\alpha\right] &= \mathbb{E}_{\xi,\boldsymbol{e}}\left[\left\|\frac{d}{\tau} f(x + \tau e, \xi)\boldsymbol{e}\right\|_2^\alpha\right] \\ &= \left(\frac{d}{\tau}\right)^\alpha \mathbb{E}_{\xi,\boldsymbol{e}}\left[\|f(x + \tau e, \xi)\|_2^\alpha \|\boldsymbol{e}\|_2^\alpha\right] \\ &\leq \left(\frac{d}{\tau}\right)^\alpha \mathbb{E}_{\xi,\boldsymbol{e}}\left[\|f(x + \tau e, \xi)\|_2^\alpha\right] \mathbb{E}_{\boldsymbol{e}}\left[\|\boldsymbol{e}\|_2^\alpha\right] \\ &\overset{\|\boldsymbol{e}\|_2=1}{=} \left(\frac{d}{\tau}\right)^\alpha \mathbb{E}_{\xi,\boldsymbol{e}}\left[\|f(x + \tau e, \xi)\|_2^\alpha\right] \\ &\overset{\text{Ass.2.7}}{\leq} \left(\frac{dG}{\tau}\right)^\alpha.\end{aligned} \tag{38}$$

2.2 For two-point feedback, the estimate was proved via a corollary of the concentration lemma for Lipschitz functions in [26]:

$$\mathbb{E}_{\xi,\boldsymbol{e}}\left[\|\boldsymbol{g}(x, \xi, \boldsymbol{e})\|_2^\alpha\right] \leq \left(\frac{\sqrt{d}M_2}{2^{1/4}}\right)^\alpha. \tag{39}$$

□

B CONVERGENCE ANALYSIS

In this section, we provide a complete formulation of the convergence theorems, presented in the paper. The proofs follow the same steps as in the one proposed in [35], using the properties of smoothed objective function and unbiasedness/boundedness of introduced gradient approximation schemes. It is also important to note that although convergence of methods is considered on bounded sets ($B_{kR}(x^*)$, where $k = 3$ or 4), an assumption on extended smoothness should be applied.

B.1 MISSING PROOFS FOR ZO-CLIPPED-SSTM-OPF

Algorithm 1: ZO-clipped-SSTM-OPF

Input: $x^0 \in Q$ — starting point
K — iteration number
B — batch size
a — step size
τ — smoothing parameter
$\{\lambda_k\}_{k=0}^{K-1}$ — clipping parameters

0.1 Set $y^0 = z^0 = x^0$;

0.2 Set starting parameters $A_0 = \alpha_0 = 0$, $L = \sqrt{d}M_2/\tau$;

for $k = 0$ **to** $K - 1$ **do**

1. Update parameters $\alpha_{k+1} = {k+2}/{2\gamma L}$, $A_{k+1} = A_k + \alpha_{k+1}$;
2. $x^{k+1} = \frac{A_k y^k + \alpha_{k+1} z^k}{A_{k+1}}$;
3. Sample $\left\{\xi_i^k\right\}_{i=1}^B \sim \mathcal{D}$ and $\left\{\boldsymbol{e}_i^k\right\}_{i=1}^B \sim \mathcal{U}(S_2^d(1))$ independently;
4. Compute $\boldsymbol{g}^B\left(x, \{\xi_i^k\}_{i=1}^B, \{\boldsymbol{e}_i^k\}_{i=1}^B\right)$ via equation 15;
5. Apply clipping $\boldsymbol{g}^{k+1} = \text{clip}\left(\boldsymbol{g}^B\left(x, \{\xi_i^k\}_{i=1}^B, \{\boldsymbol{e}_i^k\}_{i=1}^B\right)\right)$ from equation 19;
6. $z^{k+1} = z^k - \alpha^{k+1}\boldsymbol{g}^{k+1}$;
7. $y^{k+1} = \frac{A_k y^k + \alpha_{k+1} z^{k+1}}{A_{k+1}}$;

end

Output: y^K

Theorem B.1 (Full version of Theorem 3.1)**.** *Let Assumptions 2.3, 2.11 and 2.1 are held with parameter $\mu = 0$ on $B_{3R}(x^*)$. Then for $\beta \in (0, 1]$ and $K > 0$ such that $\ln {4K}/{\beta} \geq 1$ and following parameter selection*

$$a \geq \max\left\{48600 \ln^2\left(\frac{4K}{\beta}\right), \frac{1800dG(K+1)K^{\frac{1}{\alpha}} \ln^{\frac{\alpha-1}{\alpha}} \frac{4K}{\beta}}{\tau LRB^{\frac{\alpha-1}{\alpha}}}\right\}, \tag{40}$$

$$\lambda_k = \frac{R}{30\alpha_{k+1} \ln \frac{4K}{\beta}}, \tag{41}$$

after K iterations the Algorithm 1 satisfies with probability at least $1 - \beta$

$$f(y^K) - f(x^*) \leq \tau M_2 + \frac{6aLR^2}{K(K+3)}. \tag{42}$$

Moreover, with the same probability all values $\left\{x^k, y^k, z^k\right\}_{k=0}^K$ remain in ball $B_{2R}(x^*)$. In particular, when in inequality 40 an equality is achieved, then the output of ZO-clipped-SSTM satisfies

$$f(y^K) - f(x^*) \leq 2M_2\tau + \mathcal{O}\left(\max\left\{\frac{LR^2 \ln^2\left(\frac{4K}{\beta}\right)}{K^2}, \frac{dGR \ln^{\frac{\alpha-1}{\alpha}} \frac{4K}{\beta}}{\tau(BK)^{\frac{\alpha-1}{\alpha}}}\right\}\right), \tag{43}$$

meaning that to achieve desired accuracy $f(y^K) - f(x^*) < \varepsilon$ *with probability at least* $1 - \beta$ *and* $\tau = \varepsilon/4M_2$, $L = dM_2/\tau$ *algorithm* `ZO-clipped-SSTM` *requires*

$$K = \tilde{\mathcal{O}}\left(\max\left\{\frac{d^{1/4}M_2R}{\varepsilon}, \frac{1}{B}\left(\frac{dGM_2R}{\varepsilon^2}\right)^{\frac{\alpha}{\alpha-1}}\right\}\right) \text{ iterations.} \tag{44}$$

Proof. We use the result obtained in Theorem F.1 [35] for the smoothed function $\hat{f}_\tau(x)$ and one-point feedback approximation scheme. The additional randomisation used in $\hat{f}_\tau(x)$ does not affect the convergence result. Then according to it, with probability at least $1 - \beta$ after $K > 0$ iterations of Algorithm 1 all values of $\left\{x^k, y^k, z^k\right\}_{k=0}^K$ are inside the region $B_{2R}(x^*)$ and

$$\hat{f}_\tau(y^K) - \hat{f}_\tau(x^*) \leq \frac{6aLR^2}{K(K+3)}. \tag{45}$$

Applying the property from the Lemma 2.4 we obtain

$$f(y^K) - f(x^*) \leq \tau M_2 + \frac{6aLR^2}{K(K+3)}. \tag{46}$$

After substituting parameter a into equation 42 we get

$$\begin{aligned} f(y^K) - f(x^*) &\leq 2\tau M_2 + \frac{6aR^2}{K(K+3)} \\ &= 2\tau M_2 + \mathcal{O}\left(\max\left\{\frac{LR^2\ln^2\left(\frac{4K}{\beta}\right)}{K(K+3)}, \frac{dGR^2(K+1)K^{\frac{1}{\alpha}}\ln^{\frac{\alpha-1}{\alpha}}\frac{4K}{\beta}}{\tau(K+3)B^{\frac{\alpha-1}{\alpha}}}\right\}\right) \\ &\overset{\alpha>1}{=} 2\tau M_2 + \mathcal{O}\left(\max\left\{\frac{LR^2\ln^2\left(\frac{4K}{\beta}\right)}{K^2}, \frac{dGR\ln^{\frac{\alpha-1}{\alpha}}\frac{4K}{\beta}}{\tau(BK)^{\frac{\alpha-1}{\alpha}}}\right\}\right). \end{aligned} \tag{47}$$

In order to achieve accuracy ε with probability at least $1 - \beta$, we need to choose K such that each term of the maximum in equation 47, given the optimal parameters, is asymptotically equal to $\mathcal{O}(\varepsilon)$. Finally we obtain

$$K = \mathcal{O}\left(\max\left\{\frac{d^{1/4}M_2R}{\varepsilon}\ln\frac{\sqrt{d}M_2^2R^2}{\varepsilon^2\beta}, \frac{1}{B}\left(\frac{dGM_2R}{\varepsilon^2}\right)^{\frac{\alpha}{\alpha-1}}\ln\left(\frac{1}{B\beta}\left(\frac{dGM_2R}{\varepsilon^2}\right)^{\frac{\alpha}{\alpha-1}}\right)\right\}\right).$$

□

B.2 Missing proofs for R-ZO-clipped-SSTM-OPF

Algorithm 2: `R-ZO-clipped-SSTM-OPF`

Input: $x^0 \in Q$ — starting point
N — number of restarts
$\{K_t\}_{t=1}^N$ — number of iterations per restart
$\{B_t\}_{t=1}^N$ — batch size per restart
$\{a_t\}_{t=1}^N$ — step size of Algorithm 1 per restart
$\{\tau_t\}_{t=1}^N$ — smoothing parameters
$\left\{\lambda_k^1\right\}_{k=0}^{K_1-1}, \ldots, \left\{\lambda_k^N\right\}_{k=0}^{K_N-1}$ — clipping parameters

Set $\hat{x}^0 = x^0$;
for $k = 1$ **to** N **do**
  $\tilde{x}^t = \texttt{ZO-clipped-SSTM-OPF}\left(\hat{x}^k, K_t, B_t, \gamma_t, a_t, \{\lambda_k^t\}_{k=0}^{K_t-1}\right)$;
end
Output: $\tilde{x}^N$

Theorem B.2 (Full version of Theorem 3.4)**.** *Let Assumptions 2.3, 2.11 and 2.1 are held with parameter $\mu > 0$ on $B_{3R}(x^*)$. Let also $N = \left\lceil \log_2 \frac{\mu R^2}{2\varepsilon} \right\rceil$ be the number of restarts. Then at each restart $t = 1, \ldots, N$ of* `R-ZO-clipped-SSTM-OPF` *the algorithm* `ZO-clipped-SSTM-OPF` *is run with batch size B_t (chosen via equation 21) for*

$$K_t = \left\lceil \max\left\{ 1080\sqrt{\frac{L_t R_{t-1}^2}{\varepsilon_t}} \ln \frac{2160\sqrt{L_t R_{t-1}^2} N}{\sqrt{\varepsilon_t \beta}}, \frac{2}{B_t}\left(\frac{41200 dG M_2 R_{t-1}}{\varepsilon_t^2}\right)^{\frac{\alpha}{\alpha-1}} \ln\left(\frac{4N}{B_t\beta}\left(\frac{21600 dG M_2 R_{t-1}}{\varepsilon_t^2}\right)^{\frac{\alpha}{\alpha-1}}\right)\right\}\right\rceil \tag{48}$$

with the following parameter values

$$a_t \geq \max\left\{ 48600 \ln^2\left(\frac{4K_t}{\beta}\right), \frac{1800 dG(K_t+1)K_t^{\frac{1}{\alpha}} \ln^{\frac{\alpha-1}{\alpha}} \frac{4K_t}{\beta}}{\tau_t B_t^{\frac{\alpha-1}{\alpha}} L_t R_t} \right\}, \tag{49}$$

$$\lambda_k^t = \frac{R_t}{30\alpha_{k+1}^t \ln \frac{4K_t}{\beta}}, \qquad L_t = \frac{\sqrt{d}M_2}{\tau_t}, \qquad \tau_t = \frac{\varepsilon_t}{4M_2}, \tag{50}$$

$$\varepsilon_t = \frac{\mu R_{t-1}^2}{4}, \qquad R_{t-1} = \frac{R}{2^{(t-1)/2}}, \qquad A_t = \ln\left(\frac{4K_t N}{\beta}\right) \geq 1. \tag{51}$$

To achieve the accuracy ε with probability at least $1-\beta$ (with $\beta \in (0,1]$) the algorithm `R-ZO-clipped-SSTM` *requires*

$$\sum_{t=1}^{N} K_t B_t = \tilde{\mathcal{O}}\left(\max\left\{\frac{d^{1/4}M_2 R}{\sqrt{\mu\varepsilon}}, \left(\frac{dGM_2}{\sqrt{\mu\varepsilon^3}}\right)^{\frac{\alpha}{\alpha-1}}\right\}\right) \text{ oracle calls.} \tag{52}$$

With the same probability all values after the t-th restart will remain in ball $B_{2R_{t-1}}(x^*)$.

Proof. The convergence proof of the `R-clipped-SSTM` method is presented for Theorem F.2 in [35]. We use the results of equation 44 and equation 21 for each restart in calculating the number of oracle calls.

$$\sum_{t=1}^{N} K_t B_t = \sum_{t=1}^{N} \mathcal{O}\left(\left\{\frac{d^{1/4}M_2 R_{t-1}}{\varepsilon_t}\ln\left(\frac{d^{1/4}M_2 R_{t-1}\sqrt{N}}{\varepsilon_t\beta}\right), \left(\frac{\sigma R_{t-1}}{\sqrt{\varepsilon_t}}\right)^{\frac{\alpha}{\alpha-1}} \ln\left(\frac{N}{B\beta}\left(\frac{\sigma R_{t-1}}{\varepsilon_t}\right)^{\frac{\alpha}{\alpha-1}}\right)\right\}\right) \tag{53}$$

For the sake of brevity, let's denote the first term in the maximum as ①, the second as ②, and analyse each of them separately.

$$\begin{aligned}
① &= \sum_{t=1}^{N} \mathcal{O}\left(\max\left\{\frac{d^{1/4}M_2 R_{t-1}}{\varepsilon_t}\ln\left(\frac{d^{1/4}M_2 R_{t-1}\sqrt{N}}{\varepsilon_t\beta}\right)\right\}\right) \\
&= \sum_{t=1}^{N} \mathcal{O}\left(\max\left\{\frac{d^{1/4}M_2}{R_{t-1}\mu_t}\ln\left(\frac{d^{1/4}M_2\sqrt{N}}{R_{t-1}\mu_t\beta}\right)\right\}\right) \\
&= \mathcal{O}\left(\max\left\{\sum_{t=1}^{N} 2^{t/2}\frac{d^{1/4}M_2}{R\mu}\ln\left(\frac{d^{1/4}M_2\sqrt{N}}{R\mu\beta}\right)\right\}\right) \\
&= \mathcal{O}\left(\max\left\{\frac{d^{1/4}M_2}{R\mu}\ln\left(\frac{d^{1/4}M_2\sqrt{N}}{R\mu\beta}\ln\left(\frac{\mu R^2}{\varepsilon}\right)\right)\right\}\right) \\
&= \mathcal{O}\left(\max\left\{\frac{d^{1/4}M_2}{\sqrt{\mu\varepsilon}}\ln\left(\frac{d^{1/4}M_2\sqrt{N}}{\sqrt{\mu\varepsilon}\beta}\ln\left(\frac{\mu R^2}{\varepsilon}\right)\right)\right\}\right) \\
&= \tilde{\mathcal{O}}\left(\max\left\{\frac{d^{1/4}M_2}{\sqrt{\mu\varepsilon}}\right\}\right).
\end{aligned} \tag{54}$$

$$
\begin{aligned}
② &= \sum_{t=1}^{N} \mathcal{O}\left(\max\left\{\left(\frac{\sigma R_{t-1}}{\sqrt{\varepsilon_t}}\right)^{\frac{\alpha}{\alpha-1}} \ln\left(\frac{N}{B\beta}\left(\frac{\sigma R_{t-1}}{\varepsilon_t}\right)^{\frac{\alpha}{\alpha-1}}\right)\right\}\right) \\
&= \sum_{t=1}^{N} \mathcal{O}\left(\max\left\{\left(\frac{\sigma}{\mu R_{t-1}}\right)^{\frac{\alpha}{\alpha-1}} \ln\left(\frac{N}{B\beta}\left(\frac{\sigma}{\mu R_{t-1}}\right)^{\frac{\alpha}{\alpha-1}}\right)\right\}\right) \\
&= \sum_{t=1}^{N} \mathcal{O}\left(\max\left\{\left(\frac{\sigma}{\mu R_{t-1}}\right)^{\frac{\alpha}{\alpha-1}} \ln\left(\frac{N}{B\beta}\left(\frac{\sigma}{\mu R_{t-1}}\right)^{\frac{\alpha}{\alpha-1}}\right)\right\}\right) \\
&= \mathcal{O}\left(\max\left\{\left(\frac{\sigma}{\mu R}\right)^{\frac{\alpha}{\alpha-1}} \ln\left(\frac{N}{B\beta}\left(\frac{\sigma}{\mu R}\right)^{\frac{\alpha}{\alpha-1}}\right)\right\}\right) \\
&= \mathcal{O}\left(\max\left\{\left(\frac{\sigma}{\sqrt{\mu\varepsilon}}\right)^{\frac{\alpha}{\alpha-1}} \ln\left(\frac{N}{B\beta}\left(\frac{\sigma}{\sqrt{\mu\varepsilon}}\right)^{\frac{\alpha}{\alpha-1}} \ln\left(\frac{\mu R^2}{\varepsilon}\right)\right)\right\}\right) \\
&= \tilde{\mathcal{O}}\left(\left\{\left(\frac{\sigma}{\sqrt{\mu\varepsilon}}\right)^{\frac{\alpha}{\alpha-1}}\right\}\right). \qquad (55)
\end{aligned}
$$

Substituting ①, ②, and $\sigma = {}^{4dGM_2}/_{\varepsilon}$, we conclude the proof:

$$
\sum_{t=1}^{N} K_t B_t = \tilde{\mathcal{O}}\left(\max\left\{\frac{d^{1/4}M_2 R}{\sqrt{\mu\varepsilon}}, \left(\frac{dGM_2}{\sqrt{\mu\varepsilon^3}}\right)^{\frac{\alpha}{\alpha-1}}\right\}\right)
$$

□

B.3 Missing proofs for ZO-clipped-SEG

Algorithm 3: ZO-clipped-SEG

Input **Input:**

$z^0 \in \mathcal{Z}$	—	starting point
K	—	number of iterations
γ	—	step of the algorithm
τ	—	smoothing parameter
$\{\lambda_k\}_{k=0}^{K-1}$	—	clipping parameters

for $k = 0$ **to** K **do**

1. Compute $F(z^k, \xi_1^k, \boldsymbol{e}_1^k)$, where $\xi_1^k \sim \mathcal{D}_k$, $\boldsymbol{e}_1^k \sim S_2^d(1)$ sampled independently;
2. Apply clipping $\tilde{F}(z^k, \xi_1^k, \boldsymbol{e}_1^k) = \text{clip}\left(F(z^k, \xi_1^k, \boldsymbol{e}_1^k)\right)$;
3. $\tilde{z}^k = z^k - \gamma\tilde{F}(z^k, \xi_1^k, \boldsymbol{e}_1^k)$;
4. Compute $F(\tilde{z}^k, \xi_2^k, \boldsymbol{e}_2^k)$, where $\xi_2^k \sim \mathcal{D}_k$, $\boldsymbol{e}_2^k \sim S_2^d(1)$ sampled independently;
5. Apply clipping $\tilde{F}(\tilde{z}^k, \xi_2^k, \boldsymbol{e}_2^k) = \text{clip}\left(F(\tilde{z}^k, \xi_2^k, \boldsymbol{e}_2^k)\right)$;
6. $z^k = \tilde{z}^k - \gamma\tilde{F}(\tilde{z}^k, \xi_2^k, \boldsymbol{e}_2^k)$;

end

Output: $\tilde{z}_{\text{avg}}^K = \frac{1}{K+1}\sum\limits_{k=0}^{K} \tilde{z}^k$

Theorem B.3 (Full version of theorem 3.6)**.** *Let Assumptions 2.3, 2.11 and 2.1 be held on the set* $B_{4R}(z^*)$ *and*

$$
0 < \gamma \leq \min\left\{\frac{1}{160L\ln\frac{6(K+1)}{\beta}}, \frac{20^{\frac{2-\alpha}{\alpha}}R}{10800^{\frac{1}{\alpha}}(K+1)^{\frac{1}{\alpha}}\sigma\ln\frac{\alpha-1}{\alpha}\frac{6(K+1)}{\beta}}\right\}, \qquad (56)
$$

$$
\lambda_k \equiv \lambda = \frac{R}{20\gamma\ln\frac{6(K+1)}{\beta}}, \qquad \tau = \frac{\varepsilon}{4M_2}, \qquad L = \frac{\sqrt{d}M_2}{\tau}, \qquad (57)
$$

for some $\beta \in (0, 1]$ *and* $K > 0$ *such that* $\ln 6K/\beta \geq 1$. *Then after* K *iterations the iterates produced by* `ZO-clipped-SEG` *with probability at least* $1 - \beta$ *satisfy*

$$\texttt{Gap}^{SPP}_{B_R(z^*)}(\tilde{z}_{avg}) \leq 2M_2\tau + \frac{9R^2}{2\gamma(K+1)} \text{ and } \{z_k\}_{k=1}^K \in B_{3R}(z^*), \{\tilde{z}_k\}_{k=1}^K \in B_{3R}(z^*). \quad (58)$$

In particular, when γ equals the minimum from equation 56, then to achieve the desired accuracy ε with probability at least $1 - \beta$ `ZO-clipped-SEG` requires

$$K = \mathcal{O}\left(\max\left\{\frac{\sqrt{d}M_2^2R^2}{\varepsilon^2}\ln\frac{\sqrt{d}M_2^2R^2}{\varepsilon^2\beta}, \left(\frac{\sigma R}{\varepsilon}\right)^{\frac{\alpha}{\alpha-1}}\ln\frac{\sigma R}{\varepsilon\beta}\right\}\right) \text{ iterations.} \quad (59)$$

Proof. Let us use the result of Theorem G.2 [35] for the smoothed function $\hat{f}(z)$ and the approximation of the operator F. Then according to it, with probability at least $1-\beta$ after $K > 0$ iterations of Algorithm 3

$$\texttt{Gap}^{\text{VIP}}_R(\tilde{z}^K_{\text{avg}}) \leq \frac{9R^2}{2\gamma(K+1)} \quad (60)$$

for the smoothed function $\hat{f}(z)$. Moreover $\{z_k\}_{k=1}^K$ remains in the ball $B_{3R}(z^*)$ and $\{\tilde{z}_k\}_{k=1}^K$ in the ball $B_{3R}(z^*)$. Using the relation between gap metrics (see equation 25) and the result of Lemma 2.4, the following holds:

$$\texttt{Gap}^{\text{SPP}}_R(\tilde{z}^K_{\text{avg}}) \leq 2M_2\tau + \frac{9R^2}{2\gamma(K+1)}. \quad (61)$$

Thus, for non-smooth function the next inequality is fulfilled:

$$\texttt{Gap}^{\text{SPP}}_R(\tilde{z}^K_{\text{avg}}) \leq 2M_2\tau + \mathcal{O}\left(\max\left\{\frac{LR^2\ln\frac{K}{\beta}}{K}, \frac{\sigma R\ln^{\frac{\alpha-1}{\alpha}}\frac{K}{\beta}}{K^{\frac{\alpha-1}{\alpha}}}\right\}\right). \quad (62)$$

Then in order to achieve accuracy ε with probability at least $1 - \beta$ Algorithm 3 requires

$$K = \mathcal{O}\left(\max\left\{\frac{\sqrt{d}M_2^2R^2}{\varepsilon^2}\ln\frac{\sqrt{d}M_2^2R^2}{\varepsilon^2\beta}, \left(\frac{\sigma R}{\varepsilon}\right)^{\frac{\alpha}{\alpha-1}}\ln\frac{\sigma R}{\varepsilon\beta}\right\}\right) \text{ iterations.} \quad (63)$$

□

Corollary B.4. . *Let us substitute the* σ *values obtained in Lemma 2.8 for the schemes OPF and TPF operator* F *approximation schemes, respectively, into the result of Theorem 3.6.*

1. *For the one-point approximation of the* F *operator:*

$$K = \tilde{\mathcal{O}}\left(\max\left\{\frac{\sqrt{d}M_2^2R^2}{\varepsilon^2}, \left(\frac{dGM_2R}{\varepsilon^2}\right)^{\frac{\alpha}{\alpha-1}}\right\}\right); \quad (64)$$

2. *For the two-point approximation of the* F *operator:*

$$K = \tilde{\mathcal{O}}\left(\max\left\{\frac{\sqrt{d}M_2^2R^2}{\varepsilon^2}, \left(\frac{\sqrt{d}M_2R}{\varepsilon}\right)^{\frac{\alpha}{\alpha-1}}\right\}\right). \quad (65)$$

C MISSING PROOFS FOR NOISY ORACLE

For the sake of simplicity, we will introduce noisy gradient approximations schemes:

Noisy one-point feedback:

$$.\boldsymbol{g}(x, \xi, \boldsymbol{e}) = \frac{d}{\tau}f_\delta(x + \tau\boldsymbol{e}, \xi)\boldsymbol{e}; \quad (66)$$

Noisy two-point feedback:

$$.\boldsymbol{g}(x,\xi,\boldsymbol{e}) = \frac{d}{2\tau}\left(f_\delta(x+\tau\boldsymbol{e},\xi) - f_\delta(x-\tau\boldsymbol{e},\xi)\right)\boldsymbol{e}. \tag{67}$$

In our analysis we will use supplementary lemma, which estimates the influence caused by the deterministic noise on bias and variance of gradient schemes.

Lemma C.1 (Noise accumulation)**.** *If the gradient is computed using the schemes equation 66 or equation 67, then the for all $r \in \mathbb{R}^d$ holds:*

$$\mathbb{E}_{\xi,\boldsymbol{e}}\left[\langle \boldsymbol{g}(x,\xi,\boldsymbol{e}),\, r\rangle\right] \geq \left\langle \nabla \hat{f}_\tau(x),\, r\right\rangle - \frac{d\Delta\mathbb{E}_{\boldsymbol{e}}\left[\langle \boldsymbol{e},\, r\rangle\right]}{\tau}. \tag{68}$$

Moreover, an extra factor appears in α-moment estimation:

$$\textit{Noisy OPF:} \qquad \mathbb{E}_{\xi,\boldsymbol{e}}\left[\|\boldsymbol{g}(x,\xi,\boldsymbol{e})\|_2^\alpha\right] \leq \frac{1}{2}\left(\frac{2d}{\tau}\right)^\alpha (G^\alpha + \Delta^\alpha), \tag{69}$$

$$\textit{Noisy TPF:} \qquad \mathbb{E}_{\xi,\boldsymbol{e}}\left[\|\boldsymbol{g}(x,\xi,\boldsymbol{e})\|_2^\alpha\right] \leq 2^{\alpha-1}\left[\left(\frac{\sqrt{d}M_2}{2^{1/4}}\right)^\alpha + \left(\frac{d\Delta}{\tau}\right)^\alpha\right]. \tag{70}$$

Proof. 1. Firstly, we will show this result for noisy OPF scheme.

$$\begin{aligned}\boldsymbol{g}(x,\xi,\boldsymbol{e}) &= \frac{d}{\tau} f_\delta(x+\tau\boldsymbol{e},\xi)\boldsymbol{e} \\ &= \frac{d}{\tau}(f(x+\tau\boldsymbol{e},\xi) + \delta(x+\tau\boldsymbol{e},\xi))\boldsymbol{e}.\end{aligned} \tag{71}$$

Substitute this in the left-hand side of equation 68:

$$\mathbb{E}_{\xi,\boldsymbol{e}}\left[\langle \boldsymbol{g}(x,\xi,\boldsymbol{e}),\, r\rangle\right] = \frac{d}{\tau}\left(\mathbb{E}_{\xi,\boldsymbol{e}}\left[f(x+\tau\boldsymbol{e},\xi)\boldsymbol{e}\right] + \mathbb{E}_{\xi,\boldsymbol{e}}\left[\delta(x+\tau\boldsymbol{e},\xi)\boldsymbol{e}\right]\right). \tag{72}$$

The first term can be estimated by Lemma 2.4:

$$\begin{aligned}\frac{d}{\tau}\mathbb{E}_{\xi,\boldsymbol{e}}\left[\langle f(x+\tau\boldsymbol{e},\xi)\boldsymbol{e},\, r\rangle\right] &= \frac{d}{\tau}\mathbb{E}_{\boldsymbol{e}}\left[\langle \mathbb{E}_\xi\left[f(x+\tau\boldsymbol{e},\xi)\boldsymbol{e}\right],\, r\rangle\right] \\ &= \frac{d}{\tau}\mathbb{E}_{\boldsymbol{e}}\left[\langle f(x+\tau\boldsymbol{e})\boldsymbol{e},\, r\rangle\right] \\ &= \left\langle \nabla \hat{f}_\tau(x),\, r\right\rangle.\end{aligned} \tag{73}$$

Applying the Assumption 3.8 we get the estimate of the second term.

$$\frac{d}{\tau}\mathbb{E}_{\xi,\boldsymbol{e}}\left[\langle \delta(x+\tau\boldsymbol{e},\xi)\boldsymbol{e},\, r\rangle\right] \geq -\frac{d\Delta}{\tau}\mathbb{E}_{\boldsymbol{e}}\left[\langle \boldsymbol{e},\, r\rangle\right]. \tag{74}$$

Substituting the obtained results into equation 72 we get the desired one. The same steps can be applied to the two-point approximation. The only difference is the replacement of the argument $x - \tau\boldsymbol{e}$ by $x + \tau\boldsymbol{e}$ due to the symmetry of the distribution of $\boldsymbol{e}$.

$$\begin{aligned}\mathbb{E}_{\xi,\boldsymbol{e}}\left[\langle \boldsymbol{g}(x,\xi,\boldsymbol{e}),\, r\rangle\right] &= \frac{d}{2\tau}\mathbb{E}_{\xi,\boldsymbol{e}}\left[\left(f(x+\tau\boldsymbol{e},\xi) - f(x-\tau\boldsymbol{e},\xi)\right)\boldsymbol{e}\right] \\ &\quad + \frac{d}{2\tau}\mathbb{E}_{\xi,\boldsymbol{e}}\left[\left(\delta(x+\tau\boldsymbol{e},\xi) - \delta(x-\tau\boldsymbol{e},\xi)\right)\boldsymbol{e}\right] \\ &\leq \frac{d}{\tau}\mathbb{E}_{\xi,\boldsymbol{e}}\left[f(x+\tau\boldsymbol{e},\xi)\boldsymbol{e}\right] + \frac{d}{\tau}\mathbb{E}_{\xi,\boldsymbol{e}}\left[\delta(x+\tau\boldsymbol{e},\xi)\right].\end{aligned} \tag{75}$$

2. Since the two-point gradient approximation scheme was researched in [25], we will analyse only the one-point one.

$$\begin{aligned}
\mathbb{E}_{\xi,\boldsymbol{e}}\left[\|\boldsymbol{g}(x,\xi,\boldsymbol{e})\|_2^\alpha\right] &= \left(\frac{d}{\tau}\right)^\alpha \mathbb{E}_{\xi,\boldsymbol{e}}\left[\|f_\delta(x+\tau,\boldsymbol{e},\xi)\|_2^\alpha\right] \\
&= \left(\frac{d}{\tau}\right)^\alpha \mathbb{E}_{\xi,\boldsymbol{e}}\left[\|f(x+\tau,\boldsymbol{e},\xi)+\delta(x+\tau,\xi)\|_2^\alpha\right] \\
&\leq 2^{\alpha-1}\left(\frac{d}{\tau}\right)^\alpha \left(\mathbb{E}_{\xi,\boldsymbol{e}}\left[\|f(x+\tau,\boldsymbol{e},\xi)\|_2^\alpha\right] + \mathbb{E}_{\xi,\boldsymbol{e}}\left[\|\delta(x+\tau,\xi)\|_2^\alpha\right]\right) \\
&\leq 2^{\alpha-1}\left(\frac{d}{\tau}\right)^\alpha (G^\alpha+\Delta^\alpha) \\
&\leq \frac{1}{2}\left(\frac{2d}{\tau}\right)^\alpha (G^\alpha+\Delta^\alpha). \qquad (76)
\end{aligned}$$

□

We now present an auxiliary lemma from the [13].

Lemma C.2. *Let $\boldsymbol{e}$ be a random vector from a uniform distribution on the unit Euclidean sphere $S_2^d(1)$. Then, for arbitrary $r \in \mathbb{R}^d$, the following holds*

$$\mathbb{E}_{\boldsymbol{e}}\left[\langle \boldsymbol{e},\, r\rangle\right] \leq \frac{\|r\|_2}{\sqrt{d}}. \qquad (77)$$

Lemma C.3 (Lemma 3.9)**.** *To guarantee the convergence rate presented in the Theorems 3.1, 3.4 and 3.6, the noise level Δ must fulfil the following constraints:*

$$\textit{Noisy OPF:} \qquad \Delta \lesssim \min\left\{G, \frac{\varepsilon^2}{\sqrt{d}M_2R}\right\}, \qquad (78)$$

$$\textit{Noisy TPF:} \qquad \Delta \lesssim \frac{\varepsilon^2}{\sqrt{d}M_2R}. \qquad (79)$$

Proof. To implement the results from [35], it is necessary for noisy gradient approximations to be unbiased and have bounded variance. Thus, using the results from Lemma C.2 and Lemma C.1, one can get

$$\frac{\sqrt{d}\Delta\,\|r\|_2}{\tau} \geq \mathbb{E}_{\xi,\boldsymbol{e}}\left[\left\langle \boldsymbol{g}(x,\xi,\boldsymbol{e}) - \nabla\hat{f}_\tau(x),\, r\right\rangle\right]. \qquad (80)$$

In our analysis smoothing parameter is equal to $\tau = \Theta\left(\varepsilon M_2^{-1}\right)$, so substituting in the equation 80 we get

$$\frac{\sqrt{d}\Delta M_2\,\|r\|_2}{\varepsilon} \geq \mathbb{E}_{\xi,\boldsymbol{e}}\left[\left\langle \boldsymbol{g}(x,\xi,\boldsymbol{e}) - \nabla\hat{f}_\tau(x),\, r\right\rangle\right]. \qquad (81)$$

To ensure that the bias does not accumulate at the α-th moment of the gradient approximation, it is necessary for the noise term to be asymptotically smaller than the main term obtained in Lemma 2.8.

1. For OPF we get $\Delta^\alpha \lesssim G^\alpha \implies \Delta \lesssim G$.
2. For TPF we get $\left(\frac{\sqrt{d}M_2}{2^{1/4}}\right)^\alpha \lesssim \left(\frac{d\Delta}{\tau}\right)^\alpha \implies \Delta \lesssim \frac{\varepsilon}{\sqrt{d}}$.

It is easy to notice, that the asymptotic bound for TPF is weaker, then the one obtained from bias analysis. Therefore, we obtain next constraints for Δ:

$$\text{Noisy OPF:} \qquad \Delta \lesssim \min\left\{G, \frac{\varepsilon^2}{\sqrt{d}M_2R}\right\}, \qquad (82)$$

$$\text{Noisy TPF:} \qquad \Delta \lesssim \frac{\varepsilon^2}{\sqrt{d}M_2R}. \qquad (83)$$

□

D ADDITIONAL EXPERIMENTS

Consider a classical convex-concave bilinear optimisation problem

$$\min_{x\in\mathbb{R}^{d_x}} \max_{y\in\mathbb{R}^{d_y}} x^T C y + \langle \xi, z\rangle \tag{84}$$

where ξ is a random $d_x + d_y$-dimensional vector from α-stable Levy distribution with $\alpha = 1.5$, $\Delta = 10$ and z is an embedded variable from the Definition 3.5. This problem is also known as a matrix game (see [38]). In our experiment, we set $n = 100$, $d_x = d_y = 10$.

As in the Section 4, we compare several with different approximation schemes (OPF or TPF) and with and without clipping.in order to find optimal parameter values, we set the batchsize $B = 10$, and then gridsearched γ_k in the set $\{1, 10, 100\}$ and $\tau \in \{0.1, 1, 10\}$. The best convergence had been shown for the following parameter values:

1. `ZO-SEG-OPF`: $\gamma_k \equiv 10$, $B = 10$, $\tau = 1$;
2. `ZO-SEG-TPF`: $\gamma_k \equiv 10$, $B = 10$, $\tau = 1$;
3. `ZO-clipped-SEG-OPF`: $\gamma_k \equiv 100$, $B = 10$, $\tau = 1$, $\lambda = 0.1$;
4. `ZO-clipped-SEG-TPF`: $\gamma_k \equiv 100$, $B = 10$, $\tau = 1$, $\lambda = 0.1$.

The results of the comparison can be observed in the Figure 2.

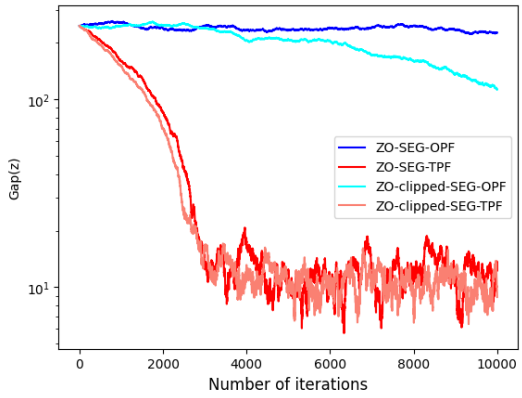

Figure 2: Comparison of the zeroth-order stochastic extrgradient methods using different gradient approximation schemes and clipping technique.

As can be seen, in the case of OPF-scheme, clipping technique offers a significant advantage, allowing the method to start converging to higher accuracy. Also in the case of two-point gradient approximation, a smaller amplitude of oscillations near the error floor is noticeable.