SIAM J. MATH. DATA SCI.
Vol. 6, No. 4, pp. 953–977

High Probability Bounds for Stochastic Subgradient Schemes with Heavy Tailed Noise*

Daniela Angela Parletta†, Andrea Paudice‡, Massimiliano Pontil§, and Saverio Salzo¶

Abstract. In this work we study high-probability bounds for stochastic subgradient methods under heavy tailed noise in Hilbert spaces. In this setting the noise is only assumed to have finite variance as opposed to a sub-Gaussian distribution for which it is known that standard subgradient methods enjoy high-probability bounds. We analyzed a clipped version of the projected stochastic subgradient method, where subgradient estimates are truncated whenever they have large norms. We show that this clipping strategy leads both to optimal *anytime* and finite horizon bounds for general averaging schemes of the iterates. We also show an application of our proposal to the case of kernel methods which gives an efficient and fully implementable algorithm for statistical supervised learning problems. Preliminary experiments are shown to support the validity of the method.

Key words. stochastic convex optimization, high-probability bounds, subgradient method, heavy tailed noise

MSC codes. 62L20, 90C25

DOI. 10.1137/22M1536558

1. Introduction.

The subgradient method was introduced in the 1960s by the Russian school of optimization as the natural generalization of the gradient descent method to non-smooth functions. It was first devised by Shor in 1962, and later studied by Polyak, Demyanov, and Ermoliev [21, 20, 24, 1]. The stochastic version of this algorithm was considered by Ermoliev in 1969 [6], who focused on the convergence of the iterates. Later on, such a method was extensively studied, with most existing results providing upper bounds on the expected optimization error in function values. Indeed, state-of-the-art convergence results ensure the optimal rate of $\mathcal{O}(1/\sqrt{k})$ for convex Lipschitz functions [19, 23]. On the other hand, high-probability bounds have proved harder to obtain. Differently from bounds in expectation, most high probability bounds have been derived under *light tails* assumptions, meaning with sub-Gaussian noise [11, 12, 14]. Recently, motivated by the fact that real world datasets are abundant but of poor quality, a line of research has started investigating high-probability bounds with *heavy-tails* assumptions, that is, with uniformly bounded variance noise. High-probability bounds in this setting have been proved in [18, 10]. Despite obtaining near-optimal

*Received by the editors November 23, 2022; accepted for publication (in revised form) May 13, 2024; published electronically October 10, 2024.
https://doi.org/10.1137/22M1536558

†CSML at Istituto Italiano di Tecnologia, 16163 Genova, Italy, and Department of Mathematics at University of Genoa, 16146 Genoa, Italy (daniela.angy@libero.it).
‡Department of Computer Science at University of Milan, 20122 Milan, Italy (and.paudice@gmail.com).
§CSML at Istituto Italiano di Tecnologia, 16163 Genova, Italy, and Department of Computer Science at UCL, London NW1 2AE, UK (massimiliano.pontil@iit.it).
¶CSML at Istituto Italiano di Tecnologia, 16163 Genova, Italy, and DIAG at Sapienza University of Rome, 00185 Rome, Italy (salzo@diag.uniroma1.it).

rates, both works suffer from either unpractical parameter settings or unrealistic assumptions. Moreover, differently from most results obtained in the light tailed case, in [18, 10] the analysis is confined to a finite horizon, which is a limitation in many practical scenarios. Indeed, finite horizon methods cannot cope with online settings in which data arrives continuously in a potentially infinite stream of batches and the predictive model is updated accordingly.

In this work we address the following optimization problem:

$$\underset{x\in X}{\text{minimize}}\ f(x), \tag{1.1}$$

where $X \subset H$ is a nonempty, closed, convex, and bounded set in a Hilbert space H with diameter $D \geq 0$, and $f\colon H \to \mathbb{R}$ is a convex Lipschitz continuous function with Lipschitz constant $L > 0$. We assume that the projection onto X can be computed explicitly but only a stochastic subgradient of f is available, that is, that for all $x \in X$, we have the following:

1. $\hat{u}(x,\xi) \in H$ and ξ is a random variable such that $\mathbb{E}[\hat{u}(x,\xi)] \in \partial f(x)$.
2. $\mathbb{E}[\|\hat{u}(x,\xi) - \mathbb{E}[\hat{u}(x,\xi)]\|^2] \leq \sigma^2$.

We stress that the only assumption made on the stochastic subgradient is that it has uniformly bounded variance, while no additional information on its distribution is available.

Contributions. In relation to problem (1.1), we study a *projected clipped stochastic subgradient method* for which we provide high-probability convergence rates under heavy tailed noise. The main contributions of this work are as follows.

- We made a new analysis of the clipped subgradient method which relies on a new decomposition of the error and different statistical properties which contrasts our analysis with that of [9, 10, 18, 13]. This allows one to obtain a first bound on the objective values which is valid for arbitrary stepsizes, weights, and clipping levels, making their role more transparent in the study of the convergence. See Theorem 2.2 and Remark 2.3.
- We provide a general convergence result when the algorithm parameters obey polynomial laws. This makes clear the setting of the stepsizes, weights, and clipping levels so to have the optimal convergence rate of $\mathcal{O}(1/\sqrt{k})$. See Theorem 2.4.
- The analysis covers both the finite-horizon and the infinite-horizon settings, general averaging schemes, and can reveal sub-Gaussian tail behavior $\mathcal{O}(\sqrt{\log(\delta^{-1})}/\sqrt{k})$ of the function values. See Corollary 2.6.
- We provide an application of the proposed method to the important case of statistical learning with kernels. Notably, the resulting algorithm is fully practicable and achieves the optimal $\mathcal{O}(1/\sqrt{k})$ rate of convergence in high probability for the excess risk.

1.1. Related work.

In this section we discuss the current literature on high-probability bounds for the stochastic nonsmooth convex setting. A summary of the state of the art is given in Table 1.[1]

Light tails. Convergence rates in high probability for light tails noise have been derived in [11, 12, 14]. In particular, in [11], under sub-Gaussian hypothesis, the last iterate of *Stochastic Gradient Descent* (SGD) was shown to achieve a convergence rate of $\mathcal{O}((\log k/\sqrt{k})\log(1/\delta))$,

[1]The rate anytime on the average iterate follows easily from [15, Proposizion 4.1]. For the readers' convenience we derive this rate in section SM3 of the supplementary material.

Table 1

Comparison of known high-probability convergence rate for nonsmooth problems. "Noise" refers to the type of oracle noise with "LT" and "HT" standing for Light Tails and Heavy Tails, respectively. "Any-Time" refers to any-time convergence guarantees. "Function Type" refers to the smoothness assumptions on f, with "Lipschitz" denoting the case of a convex Lipschitz objective, and "Composite" referring to a composite objective $f+h$ where f is convex and smooth and h is a given (nonstochastic) convex function. "Constraints" refers to the constrain set, "Bounded" refers to the optimization over a convex bounded subset of an Hilbert space.

Method	Rate	Noise	Any-Time	Function Type	Constraints	Ref.
SGD	$\frac{\log k}{\sqrt{k}}\log(1/\delta)$	LT	✓	Lipschitz	✓ (Bounded)	[11]
SGD	$\sqrt{\frac{\log(1/\delta)}{k}}$	LT	✓	Lipschitz	✓ (Bounded)	[14]
RSMD	$\sqrt{\frac{\log(1/\delta)}{k}}$	HT	✗	Composite	✓ (Bounded)	[18]
ClippedSGD	$\sqrt{\frac{\log(k/\delta)}{k}}$	HT	✗	Lipschitz	✗	[10]
ClippedSGD	$\sqrt{\frac{\log(1/\delta)}{k}}$	HT	✓	Lipschitz	✓ (Bounded)	This work

even with infinite horizon. The authors also state, without proof, that the average iterate obtains the improved rate of $\mathcal{O}(\log(1/\delta)/\sqrt{k})$, essentially matching the analogue bound in expectation. For strongly convex functions, they show that the last iterate achieves the rate $\mathcal{O}((\log k/k)\log(1/\delta))$, while the *suffix average* improves to $\mathcal{O}(1/k\log(1/\delta))$. Unfortunately, the *suffix average* can be tricky to implement in an infinite-horizon setting. Therefore, in [12] a simpler-to-implement weighted averaging scheme is shown to obtain the same rate. In [14] the previous results on the last iterate are improved to the optimal rate, but only when the time horizon is known in advance and the noise is bounded almost surely.

Heavy tails. High-probability bounds in this context have been derived for some special settings in [4, 18]. The work [4] considers a class of nonsmooth composite functions, where the objective is the sum of a smooth, strongly convex function and a general closed convex term. The authors propose an elegant and neat method, ProxBoost, that combines a robust statistical estimation procedure with the proximal point method to boost any bound in expectation into a high-probability guarantee. When used to boost the optimal method of [8], ProxBoost achieves the optimal rate of $\mathcal{O}((1/k)\sqrt{\log(1/\delta)})$. Unfortunately, this method has some practical shortcomings. First, the stochastic oracle is supposed to be available only for the smooth term of the objective function, which, in addition, is essentially limited to quadratics. Second, the proposed method requires at least three nested loops, the outer of which being the proximal point algorithm. This leads to an overall procedure which is rather convoluted and raise concerns about its practicability.

The previous composite objective structure is also analyzed in [18]. The authors remove the strong convexity assumption on the smooth part, but requires the minimization to be performed over a convex bounded domain. They develop *Robust Stochastic Mirror Descent* (RSMD), a robust version of stochastic mirror descent enjoying several desirable properties such as a near-optimal rate of $\mathcal{O}((1/\sqrt{k})\log(1/\delta))$, the absence of a batch-size and a simple constant step-size. On the other hand, the algorithm relies on the following gradient estimator:

$$\tilde{u}_k := \begin{cases} \hat{u}(x_k, \chi^k) & \text{if } \|\hat{u}(x_k, \xi^k) - \bar{g}\| \le L\|x_k - \bar{x}\| + \lambda + \nu\sigma, \\ \bar{g} & \text{otherwise.} \end{cases}$$

where $\bar{g}$ is a gradient estimate which is *close enough* to the true gradient of the smooth part of the objective at a given point $\bar{x}$. This estimator is problematic, since it is not clear how to obtain $(\bar{g}, \bar{x})$ in practice. The authors suggest generating (with high probability) $\bar{g}$ by first sampling *enough* oracle estimates at $\bar{x}$ and then by computing their geometric median; a fact that reduces the overall confidence of the procedure. However, computing such a mean estimator in high dimensions is a challenging task. Furthermore, the truncation parameter λ has to be set to

$$\lambda = \sigma\sqrt{\frac{k}{\log(1/\delta)}} + \nu\sigma,$$

which requires the knowledge of the time horizon k and does not allow for any-time guarantees. A follow-up study is [13], where the author modifies the any-time to batch conversion from [3] so to extend the guarantees of RSMD from the last average iterate to all the averages generated till the given time horizon. We note that the resulting algorithm is not fully any-time, since the initial truncation level needs to be set according to the time horizon. Differently from [18], it is assumed that the objective is smooth, while the nonsmooth case is left as an open problem. Moreover, since the gradient estimator is the same as in [18], it suffers from the same practicability issues we discussed above and the author leaves as an open problem that of identifying a more practical estimator.

Clipping strategies have already been used in [9, 10], which provide convergence rates in high probability for a fixed-horizon setting. In contrast to our work, they cover the *unconstrained* minimization of convex Lipschitz smooth functions and convex Lipschitz continuous functions, respectively. The authors analyze a clipped version of SGD, but the resulting bounds and the algorithm's parameters are subject to some limitations. In particular, the work [10], which addresses our setting, requires the algorithm parameters to be set as follows:

$$\gamma \le \min\left\{\frac{\varepsilon}{8L^2}, \frac{D}{\sqrt{2k}L}, \frac{D}{2L\log(4k/\delta)}\right\} \quad \text{step-size} \tag{1.2}$$

$$m \ge \max\left\{1, \frac{81k\sigma^2}{\lambda^2\log(4k/\delta)}\right\} \quad \text{batch-size} \tag{1.3}$$

$$\lambda = \frac{D}{\gamma\log(4k/\delta)} \quad \text{clipping-level,} \tag{1.4}$$

where k is the fixed-horizon and $\varepsilon > 0$ a free parameter. We note the restrictions in the range of the step-size γ and the batch-size m. In addition, they are coupled together. Indeed, in [10, Corollary 5.1] they show that it is possible to use $m = 1$ but at the expense of shrinking the step-size γ to $\mathcal{O}(1/\sqrt{k\log(k)})$, so that the final rate reduces to the order of $\mathcal{O}(\sqrt{\log(k/\delta)/k})$, which is not optimal in this setting. Moreover, all the parameters depend on the time horizon k, so that the convergence guarantees are not any-time. On the other hand, [9, 10] consider the minimization on unbounded domains, which is possibly more challenging than our setting.

Finally, we note that stochastic gradient methods have also been studied in conjunction with biased compressor (nonlinear) operators. See, e.g., [22] and reference therein. In this respect, we note that the clipping operator, being a projection onto a ball, is not a compressor and, moreover, it is invoked dynamically with time-varying radii.

1.2. Notation and basic facts. We set $\mathbb{N} = \{1, 2, \dots, \}$ the set of natural numbers starting from 1. We set $\mathbb{R}_+ = [0, +\infty[$ and $\mathbb{R}_{++} =]0, +\infty[$ the sets of positive and strictly positive real numbers, respectively. For any $p, q \in \mathbb{R}$ we set $p \vee q = \max\{p, q\}$, $p \wedge q = \min\{p, q\}$, and $p_+ = p \vee 0$. We denote by log the natural logarithm function. A sequence of real numbers $(\chi_k)_{k\in\mathbb{N}}$ is called increasing if, for every $k \in \mathbb{N}$, $\chi_k \leq \chi_{k+1}$. In the following, H will be a real Hilbert space and $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ will denote its scalar product and associated norm. For a convex function $f\colon H \to \mathbb{R}$, the subdifferential of f at a point $x \in H$ is defined as

$$\partial f(x) = \{u \in H \,|\, \forall\, y \in H\colon\, f(y) \geq f(x) + \langle y - x, u\rangle\}.$$

For a closed convex set $X \subset H$ we denote by P_X the orthogonal projection operator onto X.

We recall the following Bernstein's inequality for martingales [7].

Fact 1 (Freedman's inequality for martingales). *Let $(X_k)_{k\in\mathbb{N}}$ be a martingale difference sequence such that for all k,*

(i) $|X_k| \leq c$ *a.s.,*

(ii) $\sigma_k^2 := \mathbb{E}[X_k^2 | X_1, \dots, X_{k-1}] < \infty$.

Let $k \in \mathbb{N}$ and set $V_k = \sum_{i=1}^k \sigma_i^2$. Then for every η and F in $\mathbb{R}_{++}$,

$$\mathbb{P}\left(\sum_{i=1}^n X_i > \eta, V_k \leq F\right) \leq \exp\left(-\frac{1}{2}\frac{\eta^2}{F + \frac{1}{3}\eta c}\right). \tag{1.5}$$

2. The algorithm and the main results. In this section we detail the algorithm and describe the main results of this paper.

Since we are in a heavy-tails regime we consider to clip the stochastic subgradient oracle to a given level. We therefore define the following clipping operation, which corresponds to a projection onto the closed ball with radius λ:

$$(\forall\, u \in H)(\forall\, \lambda \in \mathbb{R}_{++}) \qquad \mathrm{CLIP}(u, \lambda) = \min\left\{\frac{\lambda}{\|u\|}, 1\right\} u = \begin{cases} u & \text{if } \|u\| \leq \lambda, \\ \frac{\lambda}{\|u\|} u & \text{if } \|u\| > \lambda. \end{cases}$$

The algorithm is detailed below.

Remark 2.1. In addition to the sequence x_k, Algorithm 2.1 requires keeping track of the sequences $W_k := \sum_{i=1}^k w_i$ and $\bar{x}_k$, which can be updated recursively, as $W_{k+1} = W_k + w_{k+1}$ and $\bar{x}_{k+1} = W_{k+1}^{-1}(W_k \bar{x}_k + w_{k+1} x_{k+1})$.

We will establish high-probability convergence rates in several situations depending on the choices of the weights w_k's, the stepsizes γ_k's, and the clipping levels λ_k's.

Now we are ready to provide the first main result of this paper.

Algorithm 2.1. Clipped stochastic subgradient method (C-SsGM).

Given the stepsizes $(\gamma_k)_{k\in\mathbb{N}} \in \mathbb{R}^{\mathbb{N}}_{++}$, the weights $(w_k)_{k\in\mathbb{N}} \in \mathbb{R}^{\mathbb{N}}_{++}$, the clipping levels $(\lambda_k)_{k\in\mathbb{N}} \in \mathbb{R}^{\mathbb{N}}_{++}$, the batch size $m \in \mathbb{N}$, $m \geq 1$, and an initial point $x_1 \in X$,

$$
\begin{aligned}
&\text{for } k = 1, \dots \\
&\left\lfloor
\begin{aligned}
&\text{draw } \boldsymbol{\xi}^k = (\xi_j^k)_{1\leq j\leq m} \ m \text{ independent copies of } \xi, \\
&\bar{u}_k = \frac{1}{m}\sum_{j=1}^{m} \hat{u}(x_k, \xi_j^k), \\
&\tilde{u}_k = \mathrm{CLIP}(\bar{u}_k, \lambda_k), \\
&x_{k+1} = P_X(x_k - \gamma_k \tilde{u}_k).
\end{aligned}
\right.
\end{aligned}
\tag{2.1}
$$

From the sequence $(x_k)_{k\in\mathbb{N}}$ one also defines

$$
(\forall\, k \in \mathbb{N}) \qquad \bar{x}_k = \left(\sum_{i=1}^{k} w_i\right)^{-1} \sum_{i=1}^{k} w_i x_i. \tag{2.2}
$$

Theorem 2.2 (main result 1). *Suppose that, for every $k \in \mathbb{N}$, $\lambda_k \geq (1+\varepsilon)L$ with $\varepsilon > 0$ and that the sequence $(w_k/\gamma_k)_{k\in\mathbb{N}}$ is increasing. Let $(\bar{x}_k)_{k\in\mathbb{N}}$ be the sequence generated by Algorithm 2.1. Then for every $k \in \mathbb{N}$ and $\delta \in \,]0, 2/e]$, the following holds with probability at least $1-\delta$:*

$$
\begin{aligned}
f(\bar{x}_k) - \min_X f \leq \frac{1}{\sum_{i=1}^k w_i} \Bigg[& \frac{D^2}{2}\frac{w_k}{\gamma_k} + \frac{2}{3}\left(2D \cdot \max_{1\leq i\leq k} w_i\lambda_i + \max_{1\leq i\leq k} w_i\gamma_i\lambda_i^2\right)\cdot \log\left(\frac{2}{\delta}\right) \\
& + \frac{1}{\sqrt{2}}\left(4\left(1+\frac{1}{\varepsilon}\right)\frac{D\sigma}{\sqrt{m}}\sqrt{\sum_{i=1}^k w_i^2} + \sqrt{\left(\frac{\sigma^2}{m}+L^2\right)\sum_{i=1}^k w_i^2\gamma_i^2\lambda_i^2}\right)\sqrt{\log\left(\frac{2}{\delta}\right)} \\
& + \frac{D\sigma^2}{m}\left(1+\frac{1}{\varepsilon}\right)\sum_{i=1}^k \frac{w_i}{\lambda_i} + \frac{1}{2}\left(\frac{\sigma^2}{m}+L^2\right)\sum_{i=1}^k w_i\gamma_i \Bigg].
\end{aligned}
$$

Remark 2.3. Due the above result, it is clear that in order to ensure convergence of Algorithm 2.1 we need to control the following quantities:

$$
\underbrace{\frac{w_k/\gamma_k}{\sum_{i=1}^k w_i}}_{①}, \quad \underbrace{\frac{\max_{1\leq i\leq k} w_i\lambda_i}{\sum_{i=1}^k w_i}}_{②}, \quad \underbrace{\frac{\max_{1\leq i\leq k} w_i\gamma_i\lambda_i^2}{\sum_{i=1}^k w_i}}_{③} \tag{2.3}
$$

$$
\underbrace{\frac{\sqrt{\sum_{i=1}^k w_i^2}}{\sum_{i=1}^k w_i}}_{④}, \quad \underbrace{\frac{\sqrt{\sum_{i=1}^k w_i^2\gamma_i^2\lambda_i^2}}{\sum_{i=1}^k w_i}}_{⑤}, \quad \underbrace{\frac{\sum_{i=1}^k w_i/\lambda_i}{\sum_{i=1}^k w_i}}_{⑥}, \quad \underbrace{\frac{\sum_{i=1}^k w_i\gamma_i}{\sum_{i=1}^k w_i}}_{⑦}. \tag{2.4}
$$

With the exception of term 1 which is standard in the analysis of SsGM, the rest of the quantities are related to the bias (2,4,6) and the variance (3,5,7) of the subgradient estimator.

In the rest of the section we will assume that the constants w_k, λ_k, γ_k are set as follows:

$$w_k = k^p, \quad \gamma_k = \frac{\gamma}{k^r}, \quad \lambda_k = \max\{\beta k^q, (1+\varepsilon)L\} \quad (p, r, q \in \mathbb{R}, \gamma, \beta, \varepsilon > 0) \tag{2.5}$$

and we will show conditions on the exponents so to make the quantities in (2.3) and (2.4) converging to zero. The related result is given below.

Theorem 2.4 (main result 2). *Let $(w_k)_{k\in\mathbb{N}}$, $(\gamma_k)_{k\in\mathbb{N}}$, and $(\lambda_k)_{k\in\mathbb{N}}$ be defined as in (2.5) with $p, r, q \in \mathbb{R}$ and $\beta, \varepsilon > 0$. Let $\delta \in]0, 2/e]$. Then Algorithm 2.1 converges in high probability provided that the following conditions are satisfied:*

$$p > -r, \quad r \in]0,1[, \quad q \in]0,1[, \quad \text{and} \quad q < \frac{r+1}{2}, \tag{2.6}$$

and in such case, with probability greater then $1-\delta$, the following (infinite-horizon) rate holds:

$$f(\bar{x}_k) - \min_X f = \mathcal{O}\Bigg(\frac{1}{k^{1-r}} + \frac{1}{k^{\min\{p+1,1-q\}}} + \frac{1}{k^{\min\{p+1,r+1-2q\}}} + \frac{1}{k^{\min\{p+1,\frac{1}{2}\}}} + \frac{1}{k^{\min\{p+1,r+\frac{1}{2}-q\}}} + \frac{1}{k^{\min\{p+1,q\}}} + \frac{1}{k^{\min\{p+1,r\}}} \Bigg).$$

Moreover, the optimal choices, in terms of convergence rates, for the parameters r and q are $r = 1/2$ and $q = 1/2$, and in such case the following hold.

(i) *Suppose that $p > -1/2$ and set $L_\varepsilon = (1+\varepsilon)L$. Then for every $k \in \mathbb{N}$, with probability at least $1-\delta$, we have*

$$\begin{aligned} f(\bar{x}_k) - \min_X f \le \frac{(p+1)\vee 1}{\sqrt{k}} \Bigg[& \frac{D^2}{2\gamma} \\ & + \frac{2}{3}\left(2D\max\left\{\beta, \frac{L_\varepsilon}{k^{\min\{p+\frac{1}{2},\frac{1}{2}\}}}\right\} + \gamma \max\left\{\beta^2, \frac{L_\varepsilon^2}{k^{\min\{p+\frac{1}{2},1\}}}\right\}\right)\log\frac{2}{\delta} \\ & + \frac{1}{\sqrt{2}}\left(4\left(1+\frac{1}{\varepsilon}\right)\frac{D\sigma}{\sqrt{m}}\frac{1}{((2p+1)\wedge 1)} + \gamma v_p(k)\sqrt{\frac{\sigma^2}{m}+L^2}\right)\sqrt{\log\frac{2}{\delta}} \\ & + \frac{1}{(p+1/2)\wedge 1}\left(\frac{D\sigma^2}{m}\left(1+\frac{1}{\varepsilon}\right)\frac{1}{\beta} + \frac{\gamma}{2}\left(\frac{\sigma^2}{m}+L^2\right)\right)\Bigg], \end{aligned}$$

where

$$v_p(k) = \begin{cases} \sqrt{\dfrac{2}{(2p)\wedge 1}}\max\left\{\beta, \dfrac{L_\varepsilon}{\sqrt{k}}\right\} & \text{if } p > 0, \\ \begin{cases} L_\varepsilon & \text{if } k < L_\varepsilon^2/\beta^2, \\ \beta\left(\dfrac{L_\varepsilon^2/\beta^2}{\sqrt{k}} + 1\right) & \text{if } k \ge L_\varepsilon^2/\beta^2 \end{cases} & \text{if } p = 0, \\ \dfrac{L_\varepsilon}{k^{p+\frac{1}{2}}}\sqrt{1-\dfrac{1}{2p}} + \dfrac{\max\{\beta, L_\varepsilon/\sqrt{k}\}}{\sqrt{(2p+1)\wedge 1}} & \text{if } -\frac{1}{2} < p < 0. \end{cases}$$

(ii) *Suppose that* $p = -1/2$ *and set* $L_\varepsilon = (1+\varepsilon)L$. *Then for every* $k \in \mathbb{N}$, *with probability at least* $1-\delta$, *we have*

$$f(\bar{x}_k) - \min_X f \le \frac{3}{2}\frac{1}{\sqrt{k}}\Bigg[\frac{D^2}{2\gamma} + \frac{2}{3}\left(2D\max\{\beta, L_\varepsilon\} + \gamma\max\{\beta^2, L_\varepsilon^2\}\right)\log\frac{2}{\delta} + \frac{1}{\sqrt{2}}\left(4\left(1+\frac{1}{\varepsilon}\right)\frac{D\sigma}{\sqrt{m}}(1+\log k) + \gamma u(k)\sqrt{\frac{\sigma^2}{m}+L^2}\right)\sqrt{\log\frac{2}{\delta}} + \left(\frac{D\sigma^2}{m}\left(1+\frac{1}{\varepsilon}\right)\frac{1}{\beta} + \frac{\gamma}{2}\left(\frac{\sigma^2}{m}+L^2\right)\right)(1+\log k)\Bigg],$$

where $u(k) = \sqrt{2}L_\varepsilon + \beta(1+\log k)$.

(iii) *(finite-horizon) Let* $k \in \mathbb{N}$ *and set* $(\gamma_i)_{1\le i\le k} \equiv \gamma/\sqrt{k}$, $(\lambda_i)_{1\le i\le k} \equiv (\max\{\beta\sqrt{i}, (1+\varepsilon)L\})_{1\le i\le k}$, *and* $(w_i)_{1\le i\le k} \equiv (i^p)_{1\le i\le k}$ *with* $p \ge 0$. *Then with probability at least* $1-\delta$, *we have*

$$f(\bar{x}_k) - \min_X f \le \frac{p+1}{\sqrt{k}}\Bigg[\frac{D^2}{2\gamma} + \frac{2}{3}\left(2D\max\left\{\beta, \frac{L_\varepsilon}{\sqrt{k}}\right\} + \gamma\max\left\{\beta^2, \frac{L_\varepsilon^2}{k}\right\}\right)\log\frac{2}{\delta} + \frac{1}{\sqrt{2}}\left(4\left(1+\frac{1}{\varepsilon}\right)\frac{D\sigma}{\sqrt{m}} + \gamma y_p(k)\sqrt{\frac{\sigma^2}{m}+L^2}\right)\sqrt{\log\frac{2}{\delta}} + \frac{1}{(p+1/2)\wedge 1}\frac{D\sigma^2}{m}\left(1+\frac{1}{\varepsilon}\right)\frac{1}{\beta} + \frac{\gamma}{2}\left(\frac{\sigma^2}{m}+L^2\right)\Bigg],$$

where $y_p(k) = \sqrt{2}\max\{\beta, L_\varepsilon/\sqrt{k}\}$.

Remark 2.5.

(i) The previous theorem always provides convergence (not necessarily optimal) for general parameter settings. In literature it is common to assume from the beginning that $r = q = 1/2$ [18, 10, 13]. In contrast our analysis shows a posteriori that those choices are optimal.

(ii) The previous theorem shows that for $p > -1/2$ we have an asymptotic (any time) convergence rate of $\mathcal{O}(1/\sqrt{k})$ (note that $v_p(k)$ is bounded from above) and that for $p = -1/2$ the rate degrades to $(1+\log k)/\sqrt{k}$.

(iii) Point (iii) of Theorem 2.4 provides rate in the finite horizon setting in which k (the time horizon) is fixed a priori and the stepsize is constant, set according to that time horizon. Moreover, the form of the bound allows one to optimize the stepsize. Indeed, if $k \ge (L_\varepsilon/\beta)^2$, the best stepsize (minimizing the bound) is

$$\frac{\gamma}{\sqrt{k}} = \frac{D}{\sqrt{k}}\left(\frac{4}{3}\beta^2\log\frac{2}{\delta} + 2\sqrt{\frac{\sigma^2}{m}+L^2}\sqrt{\log\frac{2}{\delta}} + \left(\frac{\sigma^2}{m}+L^2\right)\right)^{-1/2}.$$

Compared to (1.2)–(1.4) (obtained in [9, 10]), we see that in our case the batchsize m and the parameter β of the clipping levels are completely free and the rate of $\mathcal{O}(1/\sqrt{k})$ is always guaranteed. Note also that here the clipping level is not constant within the time horizon k, but it follows the policy $\lambda_i = \max\{\beta\sqrt{i}, (1+\varepsilon)L\}$.

From Theorem 2.4(i)–(iii) it is easy to derive the following result which essentially removes the log factor in the bound.

Corollary 2.6. *Let $\delta \in]0, 2/e]$ and set $\beta = \bar{\beta}/\sqrt{\log(2/\delta)}$, with $\bar{\beta} > 0$. Let $(w_k)_{k\in\mathbb{N}}$, $(\gamma_k)_{k\in\mathbb{N}}$ and $(\lambda_k)_{k\in\mathbb{N}}$ be defined as in (2.5) with $q = r = 1/2$, $p > -1/2$, and $\varepsilon > 0$. Let $(\bar{x}_k)_{k\in\mathbb{N}}$ be the sequence generated by Algorithm 2.1. Then, for every $k \geq (L_\varepsilon/\beta)^{\max\{\frac{2}{2p+1}, 2\}}$, if $p \neq 0$, and every $k \geq (L_\varepsilon/\beta)^4$, if $p = 0$, and with probability at least $1-\delta$, we have*

$$f(\bar{x}_k) - \min_X f \leq \frac{(p+1)\vee 1}{\sqrt{k}}\left[\frac{D^2}{2\gamma} + \gamma\left(\frac{2}{3}\bar{\beta}^2 + \frac{a_p\bar{\beta}}{\sqrt{2}}\sqrt{\frac{\sigma^2}{m} + L^2} + \frac{\sigma^2/m + L^2}{2((p+1/2)\wedge 1)}\right)\right.$$
$$\left. + D\left(\frac{4}{3}\bar{\beta} + \left(1+\frac{1}{\varepsilon}\right)\left(\frac{1}{(2p+1)\wedge 1}\frac{4\sigma}{\sqrt{2m}} + \frac{1}{(p+1/2)\wedge 1}\frac{\sigma^2}{m\bar{\beta}}\right)\right)\sqrt{\log\frac{2}{\delta}}\right],$$

where

$$a_p = \begin{cases} \sqrt{2((2p)\wedge 1)^{-1}} & \text{if } p > 0, \\ 2 & \text{if } p = 0, \\ \sqrt{1-(2p)^{-1}} + \sqrt{((2p+1)\wedge 1)^{-1}} & \text{if } -1/2 < p < 0. \end{cases}$$

Moreover, for finite horizon setting, with $(\gamma_i)_{1\leq i\leq k} \equiv \gamma/\sqrt{k}$, $q = 1/2$, and $p \geq 0$, we have exactly the same bound except for k that should be taken larger than $(L_\varepsilon/\beta)^2$, the constant $(p+1/2)\wedge 1$ at the denominator of the last term in the first line which is replaced by 1, and the constant a_p which now is equal to $\sqrt{2}$.

Remark 2.7.

(i) Corollary 2.6 shows that for k large enough the variable $f(\bar{x}_k) - \min_X f$ shows a sub-Gaussian tail behavior. In this respect we note that with $p \geq 1/2$ this behavior occurs earlier (with $k \geq (L_\varepsilon/\beta)^2$). This same condition on k ($k \gtrsim \log\delta^{-1}$) occurs also in other works [18, 10, 13].

(ii) The above bound allows one to optimize the initial stepsize γ, which reaches the minimum at

$$\gamma = D\left(\frac{4}{3}\bar{\beta}^2 + \sqrt{2}a_p\bar{\beta}\sqrt{\frac{\sigma^2}{m} + L^2} + \frac{\sigma^2/m + L^2}{((p+1/2)\wedge 1)}\right)^{-1/2}$$

making the final bound equal to

$$f(\bar{x}_k) - \min_X f \leq D\frac{(p+1)\vee 1}{\sqrt{k}}\left[\left(\frac{4}{3}\bar{\beta}^2 + \sqrt{2}a_p\bar{\beta}\sqrt{\frac{\sigma^2}{m} + L^2} + \frac{\sigma^2/m + L^2}{((p+1/2)\wedge 1)}\right)\right.$$
$$\left. + \left(\frac{4}{3}\bar{\beta} + \left(1+\frac{1}{\varepsilon}\right)\left(\frac{1}{((2p+1)\wedge 1)}\frac{4\sigma}{\sqrt{2m}} + \frac{1}{(p+1/2)\wedge 1}\frac{\sigma^2}{\bar{\beta}m}\right)\right)\sqrt{\log\frac{2}{\delta}}\right].$$

Again, also here the advantage of our results is clear compared to [9, 10], since the dependence on the confidence level is $\sqrt{\log(2/\delta)}$.

Remark 2.8 (on the expected optimization error). Clipping is unnecessary under our noise assumptions when in-expectation rates are in order, since the average iterate of SsGM already enjoys the optimal convergence rate

$$\mathbb{E}[f(\bar{x}_k) - f^*] \leq \sqrt{\frac{D^2(\sigma^2/m + L^2)}{k}}\,, \tag{2.7}$$

which is achieved with step-size $\gamma_i = \gamma/\sqrt{i}$ and $\gamma = \sqrt{D^2/(\sigma^2/m + L^2)}$. On the other hand, in section SM3 we could easily derive from our analysis the following in-expectation bound for the clipped-SsGM:

$$\mathbb{E}[f(\bar{x}_k) - f^*] \leq \sqrt{\frac{D^2(\sigma^2/m + L^2)}{k}} + \frac{\sigma^2}{m}\left(1 + \frac{1}{\varepsilon}\right)\sqrt{\frac{D^2}{\beta^2 k}}\,. \tag{2.8}$$

We make the following comments: (1) SsGM and C-SsGM share the same optimal step-size; (2) the rate in (2.8) is worse than that of (2.7), since it exhibits an additional term due to the bias in the subgradient estimators, a fact that we also noted in the experiments under light-tails noise; (3) finally, when $\sigma = 0$, clipping never occurs regardless of β, and C-SsGM behaves exactly as the deterministic sugradient method and the above bound reduces to

$$f(\bar{x}_k) - f^* \leq \frac{DL}{\sqrt{k}}\,, \tag{2.9}$$

which is the same rate featured by the deterministic subgradient method.

Remark 2.9 (behavior under sub-Gaussian noise). When the noise is sub-Gaussian, clipping is not necessary, and the classical SsGM method already enjoys the optimal high probability convergence rate. For example, an adaptation of the proof (reported in section SM3) of Proposition 4.1. in [15], along with the choice of $\gamma_i = \gamma/\sqrt{i}$ where

$$\gamma = \sqrt{\frac{D^2}{4(\eta^2(1 + \ln(2/\delta)) + L^2)}}$$

leads to the following high-probability convergence rate:

$$f(\bar{x}_k) - f^* \leq \frac{2D}{\sqrt{k}}\left(L + \eta\sqrt{\log\left(\frac{2e}{\delta}\right)}\right), \tag{2.10}$$

where $\eta > 0$ is the variance proxy parameter of the sub-Gaussian noise. Concerning the comparison with our bounds in Corollary 2.6, the following considerations are in order:

(i) Our bound in Corollary 2.6 (which does not take advantage of the sub-Gaussian assumption), after the optimization of the stepsize γ, shows a worse dependence on the algorithm parameters with respect to (2.10). Furthermore, (2.10) features a sub-Gaussian behavior for all k, while in the case of clipping this is only obtained for $k \gtrsim \ln(\delta^{-1})$.

(ii) Our bound depends directly on the variance σ^2, as opposed to (2.10) which depends on the variance proxy parameter η, which, in general, is larger than σ.

(iii) Clipped SsGM requires a larger number of parameters to be set with respect to SsGM. Our results provide theoretical recipes for tuning γ_i and λ_i, but, in practice, to obtain the optimal performances one has to resort to trial-and-error procedures to optimize such parameters.

(iv) In spite of the worse theoretical bounds, our experiments show that clipping SsGM may perform well even under sub-Gaussian noise, especially when the noise level is large compared to the true subgradients. See section 5.1.

3. Convergence analysis.

In this section we provide the fundamental steps of the proof of the main results of the paper. Additional details are given in sections SM1 and SM2 of the supplementary material.

3.1. Part 1.

In this section we address the proof of Theorem 2.2. We start by gathering the main statistical properties of the clipped subgradient estimator.

Lemma 3.1. *Let $x \in X$, $\lambda > L$ and define*

$$\bar{u} = \frac{1}{m}\sum_{j=1}^{m} \hat{u}(x,\xi_j) \quad \text{and} \quad \tilde{u} = \min\left\{\frac{\lambda}{\|\bar{u}\|}, 1\right\}\bar{u} = \begin{cases} \bar{u} & \text{if } \|\bar{u}\| \le \lambda \\ \frac{\lambda}{\|\bar{u}\|}\bar{u} & \text{if } \|\bar{u}\| > \lambda. \end{cases}$$

Set $u = \mathbb{E}[\hat{u}(x,\xi_j)] = \mathbb{E}[\bar{u}]$ and suppose that $\mathbb{E}[\|\hat{u}(x,\xi_j) - u\|^2] \le \sigma^2$. Then, the following hold:

(i) $\mathbb{E}\|\tilde{u}\|^2 \le \mathbb{E}\|\bar{u}\|^2 \le \frac{\sigma^2}{m} + L^2$ (second moment).

(ii) $\|\mathbb{E}\tilde{u} - u\| \le \frac{\sigma^2}{m(\lambda - L)}$ (bias).

(iii) $\mathbb{E}\|\tilde{u} - u\|^2 \le \frac{\sigma^2}{m}[1 + (\frac{\lambda+L}{\lambda-L})^2]$ (MSE).

(iv) $\mathbb{E}\|\tilde{u} - \mathbb{E}\tilde{u}\|^2 \le \frac{\sigma^2}{m}[1 + (\frac{\lambda+L}{\lambda-L})^2]$ (variance).

Remark 3.2. The previous bounds assume that $\lambda > L$. We can restate bounds (ii) and (iv) in different way. Let $\varepsilon > 0$ and suppose that $\lambda \ge L_\varepsilon$. Then we have

$$\|\mathbb{E}\tilde{u} - u\| \le \frac{\sigma^2}{m}\left(1 + \frac{1}{\varepsilon}\right)\frac{1}{\lambda} \quad \text{and} \quad \mathbb{E}\|\tilde{u} - \mathbb{E}\tilde{u}\|^2 \le \frac{\sigma^2}{m}\left[1 + \left(\frac{2+\varepsilon}{\varepsilon}\right)^2\right] \le 4\frac{\sigma^2}{m}\left(1 + \frac{1}{\varepsilon}\right)^2.$$

Now, we are ready to tackle the proof of Theorem 2.2. We start by decomposing the error in several possibly simpler terms.

Lemma 3.3 (decomposition of the error). *Let $(x_k)_{k\in\mathbb{N}}$ and $(\bar{x}_k)_{k\in\mathbb{N}}$ be generated by Algorithm 2.1. Suppose that the sequence $(w_k/\gamma_k)_{k\in\mathbb{N}}$ is increasing. Then, for all $k \in \mathbb{N}$ and $x \in X$, we have*

$$f(\bar{x}_k) - f(x) \le \frac{1}{\sum_{i=1}^k w_i}\left[\frac{D^2}{2}\frac{w_k}{\gamma_k} + \underbrace{\sum_{i=1}^k \theta_i^v}_{\mathbf{A}} + \underbrace{\sum_{i=1}^k \theta_i^b}_{\mathbf{B}} + \frac{1}{2}\underbrace{\sum_{i=1}^k \zeta_i}_{\mathbf{C}} + \frac{1}{2}\underbrace{\sum_{i=1}^k \nu_i}_{\mathbf{D}}\right] \quad a.s., \tag{3.1}$$

where

- $u_i = \mathbb{E}[\bar{u}_i \,|\, x_i] = \mathbb{E}[\hat{u}(x_i, \xi_j^i) \,|\, x_i]$ *(for* $1 \le j \le m$*),*
- $\theta_i^v := w_i \langle \tilde{u}_i - \mathbb{E}[\tilde{u}_i \,|\, x_1, \dots, x_i], x - x_i \rangle$,
- $\theta_i^b := w_i \langle \mathbb{E}[\tilde{u}_i \,|\, x_1, \dots, x_i] - u_i, x - x_i \rangle$,
- $\zeta_i := w_i \gamma_i (\|\tilde{u}_i\|^2 - \mathbb{E}[\|\tilde{u}_i\|^2 \,|\, x_1, \dots, x_i])$,
- $\nu_i := w_i \gamma_i \mathbb{E}[\|\tilde{u}_i\|^2 \,|\, x_1, \dots, x_i]$.

From Lemma 3.3 it is clear that we have to bound with high probability the four summations we named **A**, **B**, **C**, and **D**, by applying a Bernstein's type concentration inequality. So, the proof of Theorem 2.2 goes through the following steps that we collect in separate propositions.

Remark 3.4 (on the choice of the concentration inequality). Two popular tools to control bounded sums of martingale difference sequences, such as terms **A** and **C**, are Azuma–Hoeffding's and Freedman's inequalities. The former can be used to derive tight high-probability bounds for SsGM under bounded noise (and even sub-Gaussian noise). Indeed, suppose d_i is a martingale difference sequence with $|d_i| \le b_i$ almost surely. Then, Azuma–Hoeffding's inequality states that, with probability at least $1 - \delta$,

$$\sum_{i=1}^{k} d_i \lesssim \sqrt{\sum_{i=1}^{k} b_i^2 \cdot \log\left(\frac{2}{\delta}\right)}. \tag{3.2}$$

This method works well when $b_i \le$ CONSTANT as in the case of SsGM, where either $d_i = \langle \hat{u}_i - u, x - x_i \rangle$ or $d_i = \|\hat{u}_i\|^2 - \mathbb{E}[\|\hat{u}_i\|^2 | x_1, \dots, x_i]$. On the other hand, in the case of the clipped SsGM, where either $d_i = \theta_i^v$, or $d_i = \zeta_i$, we have $b_i \sim w_i \sqrt{i}$. As a result, the RHS of Azuma–Hoeffding's inequality cannot be compensated by the sum of the weights $W_k = \sum_{i=1}^k w_i$. By contrast, Freedman's inequality provides the following bound:

$$\sum_{i=1}^{k} d_i \lesssim \max_{1 \le i \le k} b_i \cdot \log\left(\frac{2}{\delta}\right) + \sqrt{F \cdot \log\left(\frac{2}{\delta}\right)}, \tag{3.3}$$

where F is the total conditional variance of the d_i's. Note that, differently from (3.2), the RHS of (3.3) only depends on the largest b_i and this ultimately can be compensated by W_k as long as $F = \mathcal{O}(W_k)$.

Proposition 3.5 (Freedman's bound). *Under the same assumptions of Fact* 1*, let* $\delta \in]0, 2/e]$*,* $k \in \mathbb{N}$*, and let* $F \ge 0$ *be s.t.* $V_k \le F$ *a.s. Then, with probability at least than* $1 - \delta/2$ *we have*

$$\sum_{i=1}^{k} X_i \le \frac{2}{3} c \log\left(\frac{2}{\delta}\right) + \sqrt{2F \log\left(\frac{2}{\delta}\right)}.$$

Proposition 3.6 (analysis of the term **A**). *Let* $k \in \mathbb{N}$ *and* $\varepsilon > 0$ *and suppose that, for every* $i = 1, \dots, k$*,* $\lambda_i \ge (1 + \varepsilon) L$*. Then, with probability at least* $1 - \delta/2$*, we have*

$$\sum_{i=1}^{k} \theta_i^v \le \frac{4}{3} D \cdot \max_{1 \le i \le k} w_i \lambda_i \cdot \log\left(\frac{2}{\delta}\right) + 2\left(1 + \frac{1}{\varepsilon}\right) \frac{D\sigma}{\sqrt{m}} \sqrt{2 \sum_{i=1}^{k} w_i^2} \cdot \sqrt{\log\left(\frac{2}{\delta}\right)}. \tag{3.4}$$

Proposition 3.7 (analysis of the term **B**). *Let $k \in \mathbb{N}$. The following hold:*

$$\sum_{i=1}^{k} \theta_i^b \le \frac{D\sigma^2}{m}\left(1+\frac{1}{\varepsilon}\right)\sum_{i=1}^{k}\frac{w_i}{\lambda_i} \qquad a.s. \tag{3.5}$$

Proposition 3.8 (analysis of the term **C**). *Let $k \in \mathbb{N}$. Then, with probability at least $1-\delta/2$,*

$$\sum_{i=1}^{k}\zeta_i \le \frac{4}{3}\cdot\max_{1\le i\le k} w_i\gamma_i\lambda_i^2\cdot\log\left(\frac{2}{\delta}\right)+\sqrt{2\left(\frac{\sigma^2}{m}+L^2\right)\sum_{i=1}^{k}w_i^2\gamma_i^2\lambda_i^2}\cdot\sqrt{\log\left(\frac{2}{\delta}\right)}.$$

The proof of the previous propositions are given in section SM1 of the supplementary material. The next one is a direct consequence of the definition of ν_i and the bound in Lemma 3.1(i).

Proposition 3.9 (analysis of the term **D**). *Let $k \in \mathbb{N}$. Then we have*

$$\sum_{i=1}^{k}\nu_i \le \left(\frac{\sigma^2}{m}+L^2\right)\sum_{i=1}^{k}w_i\gamma_i \qquad a.s.$$

Proof of Theorem 2.2. By a simple union bound, it follows from Propositions 3.6 and 3.8 that with probability at least $1-\delta$ we have

$$\sum_{i=1}^{k}\theta_i^v+\frac{1}{2}\sum_{i=1}^{k}\zeta_i \le \frac{2}{3}\left(2D\cdot\max_{1\le i\le k}w_i\lambda_i+\max_{1\le i\le k}w_i\gamma_i\lambda_i^2\right)\log\left(\frac{2}{\delta}\right)$$
$$+\frac{1}{\sqrt{2}}\left(4\left(1+\frac{1}{\varepsilon}\right)\right)\frac{D\sigma}{\sqrt{m}}\sqrt{\sum_{i=1}^{k}w_i^2}+\sqrt{\left(\frac{\sigma^2}{m}+L^2\right)\sum_{i=1}^{k}w_i^2\gamma_i^2\lambda_i^2}\Bigg)\sqrt{\log\left(\frac{2}{\delta}\right)}.$$

Since the bounds on the terms **B** and **D** hold almost surely, the statement follows directly by plugging the above bound, and those given in Propositions 3.7 and 3.9 into the inequality of Lemma 3.3. ∎

3.2. Part 2. In this section we provide the proof of Theorem 2.4. We start with the following lemmas whose proofs are provided in the section SM2 of the supplementary material.

Lemma 3.10. *Let $p \in \mathbb{R}$. Then*

$$\begin{cases}\dfrac{k^{p+1}}{(p+1)\vee 1}\le\displaystyle\sum_{i=1}^{k}i^p\le\frac{k^{p+1}}{(p+1)\wedge 1} & if\ p>-1.\\[2ex] \dfrac{1}{2}+\log k\le\displaystyle\sum_{i=1}^{k}i^p\le 1+\log k & if\ p=-1.\\[2ex] 1\le\displaystyle\sum_{i=1}^{k}i^p\le\frac{p}{p+1} & if\ p<-1.\end{cases}$$

Lemma 3.11. *Let $b,c>0$ and $t,s\in\mathbb{R}$. Then, for every integer $k\ge 1$, we have*

$$\max_{1\le i\le k}\max\{b\,i^t,c\,i^s\}=\max\{b\,k^{t_+},c\,k^{s_+}\}.$$

Lemma 3.12. *Let $b, c > 0$ and $t, s \in \mathbb{R}$ with $t > s$. Then, for every integer $k \geq 1$, we have*

$$\sum_{i=1}^{k} \max\{b\, i^t, c\, i^s\} \leq \begin{cases} \left(\frac{1}{(s+1)\wedge 1} + \frac{1}{(t+1)\wedge 1}\right) \max\left\{b, \frac{c}{k^{t-s}}\right\} \cdot k^{t+1} & \text{if } -1 < s < t, \\ \frac{k^{t+1}}{(t+1)\wedge 1} \cdot \begin{cases} c & \text{if } k < \left(\frac{c}{b}\right)^{\frac{1}{t-s}}, \\ b\left(\frac{(c/b)^2}{k^{t-s}} + 1\right) & \text{if } k \geq \left(\frac{c}{b}\right)^{\frac{1}{t-s}} \end{cases} & \text{if } -1 = s < t, \\ \frac{c\, s}{s+1} + \max\left\{b, \frac{c}{k^{t-s}}\right\} \frac{k^{t+1}}{(t+1)\wedge 1} & \text{if } s < -1 < t, \\ \frac{c\, s}{s+1} + b(1 + \log k) & \text{if } s < t = -1, \\ \frac{c\, s}{s+1} + \frac{b\, t}{t+1} & \text{if } s < t < 1. \end{cases}$$

With the help of the above results it is easy to conduct the analysis of the terms in (2.3) and (2.4). First, according to Theorem 2.2, in order to ensure that $(w_k/\gamma_k)_{k\in\mathbb{N}}$ is increasing, we have $p + r \geq 0$. Concerning the first quantity in (2.4), it follows from Lemma 3.10 that

$$④ = \frac{\sqrt{\sum_{i=1}^k w_i^2}}{\sum_{i=1}^k w_i} \leq \begin{cases} \frac{(p+1)\vee 1}{(2p+1)\wedge 1} \cdot \frac{1}{\sqrt{k}} & \text{if } -1/2 < p, \\ ((p+1)\vee 1) \cdot \frac{\sqrt{1+\log k}}{\sqrt{k}} & \text{if } p = -1/2, \\ \sqrt{\frac{2p}{2p+1}} \cdot \frac{1}{k^{p+1}} & \text{if } -1 < p < -1/2, \\ \frac{\sqrt{2}}{\log k + 1/2} & \text{if } p = -1, \\ \frac{2p}{2p+1} & \text{if } p < -1. \end{cases} \tag{3.6}$$

This implies that the left-hand side will converge to zero provided that $p \geq -1$. In particular, if $p > -1/2$, we have the best possible rate of $O(1/\sqrt{k})$. In the following we will assume $p > -1$ (since for $p = 1$ we have a slow rate, not even polynomial).

Next, we address the last two terms in (2.4). We have

$$⑦ = \frac{\sum_{i=1}^k w_i \gamma_i}{\sum_{i=1}^k w_i} \leq \gamma((p+1)\vee 1) \cdot \begin{cases} \frac{1}{(p-r+1)\wedge 1} \cdot \frac{1}{k^r} & \text{if } p - r > -1, \\ \frac{1+\log k}{k^r} & \text{if } p - r = -1, \\ \frac{p-r}{p-r+1} \cdot \frac{1}{k^{p+1}} & \text{if } p - r < -1 \end{cases} \tag{3.7}$$

and

$$⑥ = \frac{\sum_{i=1}^k w_i/\lambda_i}{\sum_{i=1}^k w_i} \leq \frac{(p+1)\vee 1}{\beta} \cdot \begin{cases} \frac{1}{(p-q+1)\wedge 1} \cdot \frac{1}{k^q} & \text{if } p - q > -1, \\ \frac{1+\log k}{k^q} & \text{if } p - q = -1, \\ \frac{p-q}{p-q+1} \cdot \frac{1}{k^{q+1}} & \text{if } p - q < -1. \end{cases} \tag{3.8}$$

Since we assumed $p > -1$, from the previous bounds, it is easy to see that, in order to have both terms converging to zero we should require in every cases that $r > 0$ and $q > 0$. Now we consider the three terms in (2.3). Concerning the first one we have

$$\textcircled{1} = \frac{w_k/\gamma_k}{\sum_{i=1}^k w_i} \le \frac{(p+1)\vee 1}{\gamma} \cdot \frac{1}{k^{1-r}}. \tag{3.9}$$

This shows that necessarily $r < 1$. Instead, for the other two quantities, recalling Lemma 3.11, we have

$$\textcircled{2} = \frac{\max_{1\le i\le k} w_i\lambda_i}{\sum_{i=1}^k w_i} \le \frac{(p+1)\vee 1}{k^{\min\{p+1,1-q\}}} \cdot \max\left\{\beta, \frac{(1+\varepsilon)L}{k^{\min\{(p+q)_+,q\}}}\right\} \tag{3.10}$$

and

$$\textcircled{3} = \frac{\max_{1\le i\le k} w_i\gamma_i\lambda_i^2}{\sum_{i=1}^k w_i} \le \frac{\gamma((p+1)\vee 1)}{k^{\min\{p+1,r+1-2q\}}} \cdot \max\left\{\beta^2, \frac{L_\varepsilon^2}{k^{\min\{(2q+p-r)_+,2q\}}}\right\}. \tag{3.11}$$

From the above bounds, since $p + 1 > 0$, it is clear that in order to have convergence to zero we should impose that

$$1 - q > 0 \quad \text{and} \quad r + 1 - 2q > 0.$$

Overall, up to now, the convergence of the considered terms is ensured if

$$p > -r, \quad r \in]0,1[, \quad q \in]0,1[, \quad \text{and} \quad q < \frac{r+1}{2}.$$

Moreover, from (3.7) and (3.9) it follows that the optimal choice of r in term of rate of convergence is $r = 1/2$. Finally, we consider the second term in (2.4). Recalling Lemma 3.12, and setting for the sake of brevity,

$$L_\varepsilon = L_\varepsilon, \quad A_{\varepsilon,\beta} = \frac{L_\varepsilon}{\beta}, \quad \text{and} \quad B_{p,r} = \frac{(2p-2r)}{2p-2r+1},$$

we have

(3.12)

$$\sum_{i=1}^k w_i^2\gamma_i^2\lambda_i^2 \le \gamma^2 \cdot \begin{cases} \dfrac{2}{(2p-2r+1)\wedge 1} \max\left\{\beta, \dfrac{L_\varepsilon}{k^q}\right\}^2 \cdot k^{2p-2r+2q+1} & \text{if } -\frac{1}{2} < p-r, \\ \dfrac{k^{2q}}{(2p-2r+2q+1)\wedge 1} \cdot \begin{cases} L_\varepsilon^2 & \text{if } k < A_{\varepsilon,\beta}^{1/q}, \\ \beta^2\left(\frac{A_{\varepsilon,\beta}^4}{k^{2q}}+1\right) & \text{if } k \ge A_{\varepsilon,\beta}^{1/q} \end{cases} & \text{if } -\frac{1}{2} = p-r, \\ L_\varepsilon^2 B_{p,r} + \max\left\{\beta, \dfrac{L_\varepsilon}{k^q}\right\}^2 \dfrac{k^{2p-2r+2q+1}}{(2p-2r+2q+1)\wedge 1} & \text{if } p-r < -\dfrac{1}{2} \\ & \quad < p-r+q, \\ L_\varepsilon^2 B_{p,r} + \beta^2(1+\log k) & \text{if } p-r+q = -\frac{1}{2}, \\ L_\varepsilon^2 B_{p,r} + \dfrac{\beta^2(2p-2r+2q)}{2p-2r+2q+1} & \text{if } p-r+q < -\frac{1}{2}, \end{cases}$$

and hence

$$\text{⑤} = \frac{\sqrt{\sum_{i=1}^k w_i^2\gamma_i^2\lambda_i^2}}{\sum_{i=1}^k w_i} \le \gamma\cdot((p+1)\vee 1)$$
$$\times \begin{cases} \sqrt{\dfrac{2}{(2p-2r+1)\wedge 1}}\max\left\{\beta, \dfrac{L_\varepsilon}{k^q}\right\}\cdot\dfrac{1}{k^{r+\frac12-q}} & \text{if } -\tfrac12 < p-r, \\ \dfrac{1}{\sqrt{(2q)\wedge 1}}\cdot\dfrac{1}{k^{r+\frac12-q}}\cdot\begin{cases}(1+\varepsilon)L & \text{if } k < A_{\varepsilon,\beta}^{1/q},\\ \beta\left[\frac{A_{\varepsilon,\beta}^2}{k^q}+1\right] & \text{if } k \ge A_{\varepsilon,\beta}^{1/q}\end{cases} & \text{if } -\tfrac12 = p-r, \\ \left(\dfrac{L_\varepsilon\sqrt{B_{p,r}}}{k^{p+q-r+\frac12}} + \max\left\{\beta,\dfrac{L_\varepsilon}{k^q}\right\}\dfrac{1}{\sqrt{(2p-2r+2q+1)\wedge 1}}\right)\dfrac{1}{k^{r+\frac12-q}} & \text{if } p-r < -\dfrac12 \\ & \quad < p-r+q, \\ (L_\varepsilon\sqrt{B_{p,r}}+\beta\sqrt{1+\log k})\cdot\dfrac{1}{k^{r+\frac12-q}} & \text{if } p-r+q = -\tfrac12, \\ \left(L_\varepsilon\sqrt{B_{p,r}}+\beta\sqrt{\dfrac{2p-2r+2q}{2p-2r+2q+1}}\right)\cdot\dfrac{1}{k^{p+1}} & \text{if } p-r+q < -\tfrac12. \end{cases}$$

This bound shows that convergence is guaranteed if $r+1/2-q>0$, that is $q<r+1/2$. Taking into account that $r>0$, this provide a weaker condition than the condition $q<(r+1)/2$ already obtained. Moreover, with the optimal choice of $r=1/2$, ⑤ features a rate of $1/k^{1-q}$, which, considering that $q>0$, is optimaized when $q=1/2$. In the end, Theorem 2.4(i)(ii) follows by simply plugging the above bounds with $q=r=1/2$ in Theorem 2.2. Finally, concerning Theorem 2.4(iii), we note that if we take the stepsizes constant till k, that is $(\gamma_i)_{1\le i\le k}\equiv\gamma$ (so that $r=0$), then we should ask for $p\ge 0$ and the bounds for ①, ③, ⑤, and ⑦ (with $q=1/2$) become

$$\text{①} \le \frac{p+1}{\gamma}\cdot\frac{1}{k}, \quad \text{③} \le \gamma(p+1)\cdot\max\left\{\beta^2, \frac{L_\varepsilon^2}{k}\right\}$$
$$\text{⑤} \le \gamma(p+1)\sqrt{2}\max\left\{\beta,\frac{L_\varepsilon}{\sqrt{k}}\right\}\frac{1}{\sqrt{k}}, \quad \text{⑦} \le \gamma(p+1).$$

Thus, if we choose $(\gamma_i)_{1\le i\le k}\equiv\gamma/\sqrt{k}$ and substitute the bounds in Theorem 2.2, the statement of Theorem 2.4(iii) follows.

4. Application to kernel methods. In this section we present an application of Algorithm 2.1 to the case of kernel methods and show that the clipping strategy can be implemented in this setting by updating finite-dimensional variables. This resembles the classical kernel trick which is common in empirical risk minimization [25] and in SGD as well [5, 17, 16], Here we adjust the method so to handle the nonlinearity of the clipping operation. In passing, we note that our weak assumption on the noise allows to treat unbounded kernels as the polynomial kernel and contrasts with the common bounded kernel assumption [2, 26].

The problem we address is formulated as follows:

$$\min_{x\in B_r} R(x) := \mathbb{E}[\ell(\langle x, \Phi(Z)\rangle, Y)],$$

where Z and Y are random variables with values in the measurable spaces $\mathcal{Z}$ and $\mathcal{Y}$, respectively, with joint distribution μ; H is a Hilbert space with scalar product $\langle\cdot,\cdot\rangle$ and B_r is the ball of radius $r>0$ centered at the origin of H. We assume the *loss function* $\ell:\mathbb{R}\times\mathcal{Y}\to\mathbb{R}_+$ to be convex and L-Lipschitz in the first argument (for every fixed value of the second argument) and we assume that $\mathbb{E}[\ell(0,Y)]<+\infty$. The function $\Phi:\mathcal{Z}\to H$ is the *feature map* and $K(\cdot,\cdot)=\langle\Phi(\cdot),\Phi(\cdot)\rangle$ is the corresponding kernel. We further assume that $\mathbb{E}[\|\Phi(Z)\|^2]<+\infty$. The goal is to find a function $\mathcal{Z}\ni z\mapsto\langle x,\Phi(z)\rangle\in\mathbb{R}$, with $x\in H$, which minimizes the expected risk above over the ball B_r, based on the possibility of sampling from the distribution μ.

Derivation. Assume $x_0=0$ and recall that, for each $k\geq 0$, the main update in Algorithm 2.1 is defined by the following:

$$x_{k+1}=P_{B_r}(x_k-\gamma_k\tilde{u}_k), \tag{4.1}$$

where $\tilde{u}_k=\mathrm{CLIP}(\bar{u}_k,\lambda_k)$, $\bar{u}_k=\frac{1}{m}\sum_{j=1}^m\hat{u}(x_k,(Z_j^k,Y_j^k))$, $\hat{u}(x,(Z,Y))=\ell'(\langle x,\Phi(Z)\rangle,Y)\Phi(Z)$, where, for all $(t,y)\in\mathbb{R}\times\mathcal{Y}$, $\ell'(t,y)$ is a subgradient of $\ell(\cdot,y)$ at t, that is, $\ell'(t,y)\in\partial\ell(t,y)$. Since H may be infinite dimensional, computing $\tilde{u}_k$ with a straightforward application of its definition may be problematic.

In order to solve the aforementioned issue, the algorithm will keep an implicit representation of the iterates x_k's in terms of the kernels. To obtain this representation, we start by noticing that since P_{B_r} is the projection in the ball centered at the origin, then

$$x_{k+1}=\min\left\{\frac{r}{\|x_k-\gamma_k\tilde{u}_k\|},1\right\}(x_k-\gamma_k\tilde{u}_k)=\delta_k(x_k-\gamma_k\tilde{u}_k), \tag{4.2}$$

where we set $\delta_k=\min\{\frac{r}{\|x_k-\gamma_k\tilde{u}_k\|},1\}$. Similarly, by using the definitions of clipping and $\hat{u}(x,(Z,Y))$ it holds that

$$\tilde{u}_k=\min\left\{\frac{\lambda_k}{\|\bar{u}_k\|},1\right\}\bar{u}_k=\frac{\rho_k}{m}\sum_{j=1}^m\hat{u}(x_k,(Z_j^k,Y_j^k))=\frac{\rho_k}{m}\sum_{j=1}^m\alpha_j^k\Phi(Z_j^k), \tag{4.3}$$

where we set $\rho_k=\min\{\frac{\lambda_k}{\|\bar{u}_k\|},1\}$ and $\alpha_j^k=\ell'(\langle x_k,Z_j^k\rangle,Y_j^k)$. From (4.3) we have that

$$\|\bar{u}_k\|^2=\frac{1}{m^2}\sum_{j,j'=1}^m\alpha_j^k\alpha_{j'}^kK(Z_j^k,Z_{j'}^k). \tag{4.4}$$

The above equation allows for the computation of ρ_k once the α_j^ks are known. Furthermore, combing (4.3) and (4.2) we obtain that

$$\begin{aligned}x_{k+1}=\delta_k(x_k-\gamma_k\tilde{u}_k)&=\delta_k\left(x_k-\gamma_k\frac{\rho_k}{n}\sum_{j=1}^n\alpha_j^k\Phi(Z_j^k)\right)=\delta_kx_k-\frac{\delta_k\gamma_k\rho_k}{n}\sum_{j=1}^n\alpha_j^k\Phi(Z_j^k)\\&=\delta_kx_k+\sum_{j=1}^n\left(-\frac{\delta_k\gamma_k\rho_k}{n}\right)\alpha_j^k\Phi(Z_j^k)=\sum_{i=0}^k\sum_{j=1}^n a_{ij}^k\Phi(Z_j^i)\,,\end{aligned} \tag{4.5}$$

where we set

$$a_{ij}^k = \begin{cases} \delta_k a_{ij}^{k-1} & \text{if } i \le k-1, \\ -\frac{\delta_k \gamma_k \rho_k}{n} \alpha_j^k & \text{if } i = k. \end{cases} \tag{4.6}$$

As a consequence,

$$\begin{aligned} \|x_k - \gamma_k \tilde{u}_k\|^2 &= \|x_k\|^2 + \gamma_k^2 \|\tilde{u}_k\|^2 - 2\gamma_k \langle x_k, \tilde{u}_k \rangle \\ &= \|x_k\|^2 + \gamma_k^2 \rho_k^2 \|\bar{u}_k\|^2 - 2\gamma_k \rho_k \langle x_k, \bar{u}_k \rangle \\ &= \|x_k\|^2 + \gamma_k^2 \rho_k^2 \|\bar{u}_k\|^2 - 2\frac{\gamma_k \rho_k}{n} \left\langle x_k, \sum_{j'=1}^n \alpha_{j'}^k \Phi(Z_{j'}^k) \right\rangle \\ &= \|x_k\|^2 + \gamma_k^2 \rho_k^2 \|\bar{u}_k\|^2 - 2\frac{\gamma_k \rho_k}{n} \sum_{i=0}^{k-1} \sum_{j=0}^n \sum_{j'=1}^n a_{ij}^{k-1} \alpha_{j'}^k K(Z_j^i, Z_{j'}^k). \end{aligned}$$

Note that the above equation allows for the computation of δ_k once the α_j^ks (and then the a_{ij}^ks) are known. Replacing the expression of x_k from (4.5) in the definition of α_j^k leads to

$$\alpha_j^k = \ell'(\langle x_k, \Phi(Z_j^k) \rangle, Y_j^k) = \ell' \left(\sum_{i=0}^{k-1} \sum_{j'=1}^m a_{ij'}^{k-1} K(Z_{j'}^i, Z_j^k), Y_j^k \right), \tag{4.7}$$

which can be computed directly using the kernels.

Algorithm. As mentioned in the previous paragraph, the algorithm keeps an implicit representation for the iterates x_k which is computed as follows. This is enough to make predictions on new points as we are going to show in the following. We notice that, at any time k, it is possible to make a prediction for an instance Z using the kth iterate as follows:

$$\langle x_{k+1}, \Phi(Z) \rangle = \delta_k \langle x_k, \Phi(Z) \rangle - \frac{\delta_k \gamma_k \rho_k}{m} \sum_{j=1}^m \alpha_j^k K(Z_j^k, Z). \tag{4.8}$$

This prediction requires nothing but the prediction made with the previous iterate, $\delta_k, \rho_k, \alpha_j^k$s and the values of the kernels $K(Z_j^k, Z)$. It is easy to observe that, by recursion, there is no need to have an explicit expression for the x_k; instead it is enough to compute, along the iterations, only $\delta_k, \rho_k, a_{ij}^k$, and α_j^k. The algorithm to compute the iterates x_k in the implicit form is given in Algorithm 4.1 (along with the initialization procedure described in Algorithm 4.2). Now, since we have developed the theory for weighted average schemes, let $(w_k)_{k \in \mathbb{N}}$ be a sequence of nonnegative weight, and define

$$W_{k+1} = \begin{cases} w_1 & \text{if } k = 0, \\ W_k + w_{k+1} & \text{otherwise.} \end{cases} \tag{4.9}$$

Then, by considering $\bar{x}_k$ as defined in Algorithm 2.1, the prediction can be computed as

$$\langle \bar{x}_{k+1}, \Phi(Z) \rangle = \frac{1}{W_{k+1}} \left(W_k \langle \bar{x}_k, \Phi(Z) \rangle + \langle x_{k+1}, \Phi(Z) \rangle \right). \tag{4.10}$$

Algorithm 4.1. Kernel C-SsGM.

Given the step-sizes $(\gamma_k)_{k\in\mathbb{N}}\in\mathbb{R}^{\mathbb{N}}_{++}$, the weights $(w_k)_{k\in\mathbb{N}}\in\mathbb{R}^{\mathbb{N}}_{++}$, the clipping levels $(\lambda_k)_{k\in\mathbb{N}}\in\mathbb{R}^{\mathbb{N}}_{++}$, the batch size $m\in\mathbb{N}$, $m\geq 1$, and an initial point $x_0=0$, do the following:

$\textsc{Initialization}\,(\gamma_0,\lambda_0,m)$
for $k=1,\dots$
- draw $(Z_j^k,Y_j^k)_{1\leq j\leq m}$ m independent copies of (Z,Y),
- pick $\alpha_j^k=\ell'\left(\sum_{i=0}^{k-1}\sum_{j'=1}^{m}a_{ij'}^{k-1}K(Z_{j'}^i,Z_j^k),Y_j^k\right)$, for each $1\leq j\leq m$,
- compute
  - $\|\bar{u}_k\|^2=\frac{1}{n^2}\sum_{j,j'=1}^{m}\alpha_j^k\alpha_{j'}^kK(Z_j^k,Z_{j'}^k)$,
  - $\rho_k=\min\left\{\frac{\lambda_k}{\|\bar{u}_k\|},1\right\}$,
  - $\|x_k-\gamma_k\tilde{u}_k\|^2=\|x_k\|^2+\gamma_k^2\rho_k^2\|\bar{u}_k\|^2-2\frac{\gamma_k\rho_k}{m}\sum_{i=0}^{k-1}\sum_{j,j'=1}^{m}a_{ij}^{k-1}\alpha_{j'}^kK(Z_j^i,Z_{j'}^k)$,
  - $\delta_k=\min\left\{\frac{r}{\gamma_k\rho_k\|x_k-\gamma_k\tilde{u}_k\|},1\right\}$,
  - $a_{ij}^k=\begin{cases}\delta_k a_{ij}^{k-1}, & \text{if } i\leq k-1,\\ -\frac{\delta_k\gamma_k\rho_k}{m}\alpha_j^k & \text{otherwise,}\end{cases}\quad 1\leq j\leq m$,
  - $\|x_{k+1}\|=\delta_k\|x_k-\gamma_k\tilde{u}_k\|$.

From the sequence $(x_k)_{k\in\mathbb{N}}$ one also defines

$$(\forall\, k\in\mathbb{N})\qquad \bar{x}_k=\left(\sum_{i=1}^{k}w_i\right)^{-1}\sum_{i=1}^{k}w_ix_i. \tag{4.11}$$

Remark 4.1. Notice that the prediction made with the $k+1$th iterate in (4.8) can be computed recursively from the prediction made by the kth iterate. Similarly, the prediction made by the $k+1$th weighted average in (4.10) can be computed by the prediction made by the previous weighted average. Both there recursion allows for significant computational saving, as at each step it only necessary to compute the kernels among Z and the instances of the kth batch.

Remark 4.2. We note that in this setting

$$\begin{aligned}\mathbb{E}[\|\hat{u}(x,(Z,Y))-\mathbb{E}[\hat{u}(x,(Z,Y))]\|^2]&\leq\mathbb{E}[\|\hat{u}(x,(Z,Y))\|^2]\\&=\mathbb{E}[|\ell'(\langle x,Z\rangle,Y)|^2\|\Phi(Z)\|^2]\\&\leq L^2\mathcal{K}\end{aligned} \tag{4.12}$$

Algorithm 4.2. INITIALIZATION.

Given the step-size γ, clipping level λ and the batch size $m \in \mathbb{N}$, $m \geq 1$,

draw $(Z_j^0, Y_j^0)_{1 \leq j \leq m}$ m independent copies of (Z, Y),

pick $\alpha_j^0 = \ell'(0, Y_j^0)$, for each $1 \leq j \leq m$,

compute

$$
\begin{aligned}
&\|\bar{u}_0\|^2 = \frac{1}{n^2} \sum_{j,j'=1}^{m} \alpha_j^0 \alpha_{j'}^0 K(Z_j^0, Z_{j'}^0), \\
&\rho_0 = \min\left\{ \frac{\lambda_0}{\|\bar{u}_0\|}, 1 \right\}, \\
&\delta_0 = \min\left\{ \frac{r}{\gamma_0 \rho_0 \|\bar{u}_0\|}, 1 \right\}, \\
&a_{0,j}^0 = -\frac{\delta_0 \gamma_0 \rho_0}{m} \alpha_j^0, 1 \leq j \leq m, \\
&\|x_1\| = \gamma_0 \delta_0 \rho_0 \|\bar{u}_0\|.
\end{aligned}
\tag{4.13}
$$

where, $\mathcal{K} := \mathbb{E}[K(Z, Z)]$ and the last line follows from the Lipschitzness of the loss. The important consequence of Equation (4.12) is that in order to obtain a convergence rate it is enough to assume that the expected value of the kernel is bounded.

5. Numerical experiments.

In this section we present four experiments on synthetic problems, in order as follows:

- a comparison between C-SsGM and standard SsGM under heavy-tailed and light-tailed noise;
- a comparison between different averaging schemes in infinite- and finite-horizon settings for C-SsGM;
- the advantages and practicability of our fully flexible parameter setting over those prescribed in [10];
- a preliminary experiment on the kernel-based method to learn a nonlinear classifier.

5.1. Comparison with standard SsGM under light and heavy tails noise.

The goal of this experiment is to compare the performance of C-SsGM against the standard SsGM, under different noise regimes.

We consider the minimization of $f(x) = |x|$ over $[-1/2, 1/2]$. The subgradient oracle returns $\text{SIGN}(x) + n$ where $\text{SIGN}(x) = 1$ for $x \leq 0$ and -1 otherwise, and n is either a Pareto random variable with shape parameter 2.1, or a Gaussian random variables, both with zero-mean and unit variance. Note that the considered Pareto noise has no moment of order greater than 2.1 and thus fits well our heavy-tailed noise assumption. We run SsGM and C-SsGM for 1000 iterations. We set the clipping level parameters β and ϵ to 0.01 and 0.001, respectively. As for the step-size, we set $\gamma_k = \gamma/\sqrt{k}$, where γ is computed via a grid search over $\{0.01, 0.05, 0.1, 0.2, 0.3, 0.5, 0.6, 1\}$. We note that among the previous values we have included the optimal values of γ for the in-expectation and the high-probability bounds as discussed

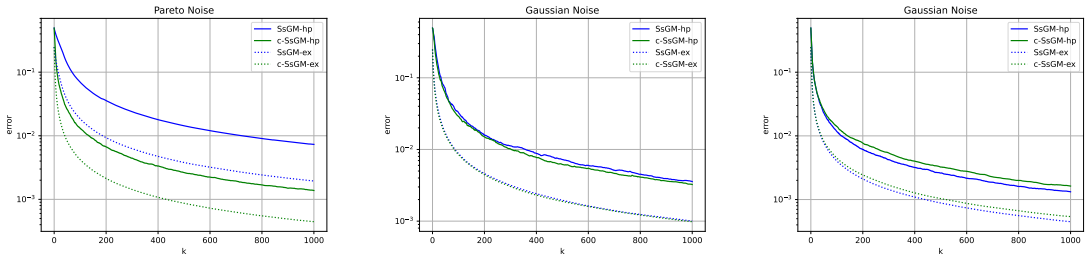

Figure 1. SsGM *versus* C-SsGM *under different types of noise: Pareto noise* (*left*)*, Gaussian noise with* $\sigma = 1$ (*center*) *and* $\sigma = 0.1$ (*right*)*. The curves show the* 99*th percentile and the average of the optimization error over* 1000 *repetitions.*

in Remarks 2.8 and 2.9, and item (ii) in Remark 2.7. For both algorithms we consider the standard weights $w_k = 1$ and batch size $m = 1$. We ran both methods 1000 times and collected the average and the 99th percentile of the optimization error. The results, reported in Figure 1, show that, under Pareto noise, the clipping strategy provides a notable acceleration of the convergence, and at the same time a significant reduction of the upper deviations from the mean (note that the y axis is in log scale). We note that this acceleration occurs even in expectation, which is not quite aligned with the theoretical bounds given in Remark 2.8. Moreover, under Gaussian noise with large variance (Figure 1, center), C-SsGM still performs well against the standard SsGM, despite the fact that the clipping is, in principle, not needed for the convergence in high probability (see Remark 2.9) and ultimately introduces bias. We believe that, since in the above cases the standard deviation of the noise is large compared to the true subgradients, the reduction of the variance obtained by the clipping strategy helps and may compensate the inherent distortion. We also tested the algorithms under a Gaussian noise with smaller variance, i.e., with $\sigma = 0.1$. In this case SsGM performs better than C-SsGM as we may expect from the theory. Finally, we note that our choice of β corresponds to a clipping level of $(1+\varepsilon)L$ across all the iterations. We checked that this aggressive schedule leads to the best performances.

5.2. Comparison of iteration averaging schemes. The goal of this experiment is to compare the performances of the different C-SsGM schemes discussed in section 2.

We consider the minimization of $f(x) = \|x\|_1$ over the ℓ_2 ball of radius 1 in $\mathbb{R}^d$, with $d = 100$, which contains the global minimum 0 corresponding to $f^* = 0$. Notice that the objective is Lipschitz continuous with $L = 10$. At each x, the oracle's answer is built by first computing a subgradient of f and then adding a zero mean noise vector $n \in \mathbb{R}^d$ with variance $\sigma^2 = 100$. The noise vector is generated by independently sampling its components from a Pareto distribution with shape parameter 2.1 with zero mean and unit variance. We test several average schemes for the iterates with $p = -1/2, p = 0$, and $p = 1/2$ in the infinite horizon setting. We also consider the finite horizon setting and the coordinatewise variant of clipping described in [27] with $p = 0$. We set the step-size schedule as $\gamma_i = \gamma/\sqrt{i}$, the clipping-level as $\lambda_i = \max\{\beta\sqrt{i}, (1+\varepsilon)L\}$, and $\varepsilon = 0.001$. We perform a grid search for the parameters γ and β in $\{0.01, 0.02, 0.03, 0.05, 0.06, 0.1, 0.2, 0.3\}$ and $\{0.32, 0.64, 1.28\}$, respectively. In the case of coordinatewise clipping, the value of β is further divided by the Lipschitz constant 10.

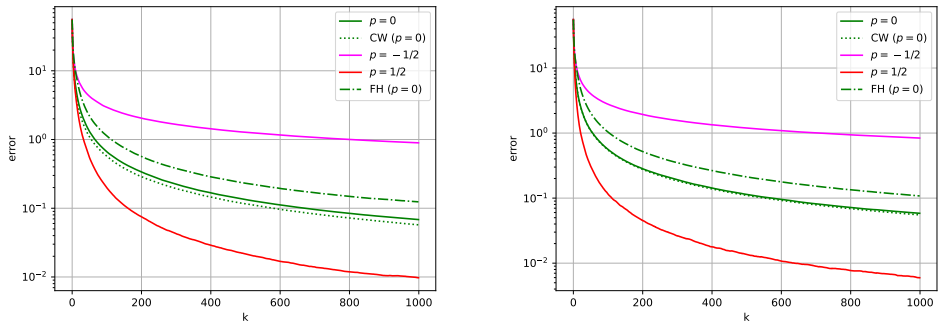

Figure 2. *Comparison of different averaging schemes for the iterations in infinite and finite-horizon settings with batch sizes $m = 1$ (left) and $m = \sigma$ (right). The curves show the 99th percentile of the optimization error over 100 repetitions.*

We run 100 times all the algorithms with batch sizes $m = 1$ and $m = \sigma$,[2] till $k = 1000$ iterations and measure the 99th percentile of the optimization error. Results are shown in Figure 2 where it is possible to see that: (1) the averaging scheme with $p = 1/2$ performs the best, while that with $p = -1/2$ is the worst (see also Remark 2.5(ii)); (2) the three schemes with $p = 0$ performs similarly, with the coordinatewise variant doing slightly better than the others.

5.3. Comparison with [10]. Here we make a comparison with the clipped-SGD algorithm from [10] on the same problem considered in section 5.2. The algorithm is essentially the same as C-SsGM except for the projection step and the different parameter setting. In this respect we note that although, in general, clipped-SGD cannot handle constraints, it can be used to solve the specific problem at hand, since in this case the constrained and the unconstrained problems share the same solution. Moreover, we consider the finite-horizon scenario with $w_k = 1$ which is the same studied in [10]. For the sake of a fair comparison we run the algorithms with the same batchsize. As for clipped-SGD in [10], due to the restrictions on the algorithm's parameters recalled in (1.2)–(1.4), by varying ε one can obtain the following explicit range for the stepsize:

$$\gamma \le \gamma_{\max} := D \cdot \min\left\{ \frac{\sqrt{m}}{9\sigma\sqrt{k\log(4k/\delta)}}, \frac{1}{\sqrt{2k}L}, \frac{1}{2L\log(4k/\delta)} \right\}. \tag{5.1}$$

We perform a grid search on γ in $\{\gamma_{\max}/4, \gamma_{\max}/2, \gamma_{\max}\}$. Whereas, for our clipped-SGM the grid search is done on the parameters $(\gamma, \beta) \in \{0.1, 0.2, 0.3\} \times \{0.32, 0.64, 1.28\}$. Note that in our algorithm with finite horizon setting the stepsize is constant and equal to $\gamma/\sqrt{k}$, while the clipping level is possibly varying as $\max\{\beta\sqrt{i}, (1+\varepsilon)L\} = \max\{\beta\sqrt{i}, 10.01\}$. See Theorem 2.4(iii).

We report in Table 2 the 99th percentile of the optimization error over 100 repetitions of the algorithms. We also indicate the best parameters chosen via the greed search procedure. As it is possible to see the parameter setting in [10] gives tiny values of the stepsizes and quite large values of the clipping levels, which ultimately leads to a noticeable worst performance.

[2]We also test the algorithms with batchsize $m = \sigma^2$ but the results are pretty similar to those with $m = \sigma$.

Table 2

Comparison of C-SsGM with clipped-SGD from [10] in the finite-horizon scenario. The parameter m denotes the batchsize. $C-SsGM$ We report the 99th-percentile of the optimization error over 100 runs of the experiment.

Method	m	Stepsize	Clipping level	99th-percentile of the opt. error
C-SsGM	1	10^{-2}	10.01	0.124
	10	10^{-2}	10.01	0.108
	100	10^{-2}	$10.01-20.02$	0.113
CLIPPED-SGD [10]	1	10^{-4}	792.4	4.720
	10	3×10^{-4}	250.6	1.570
	100	10^{-3}	79.2	0.463

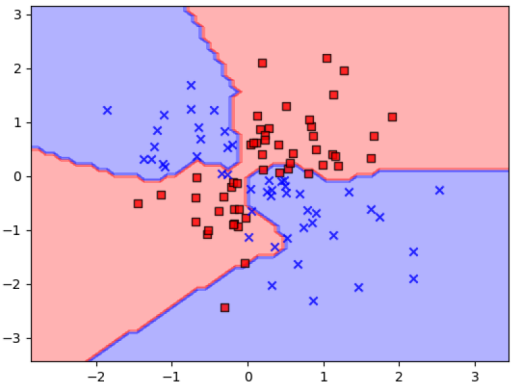

Figure 3. *Decision boundary of the classifier learned by Algorithm 4.1 with Gaussian kernel along with the negative (red squares) and the positive (blue squares) training examples.*

We make a final comment on this comparison. In the numerical section of [10], in order to overcome the narrow range of values allowed by the theory, the authors perform a grid-search on the stepsize without the limitation given in (5.1). However, in doing so there is no mathematical guarantee that the selected stepsize leads to a convergent algorithm (according to Theorem 5.1 in [10]).

5.4. Nonlinear classification. We aim at learning a classifier with a low expected classification error when the data generating distribution is a bivariate standard normal and the target function is the indicator of the first and the third orthants. As a convex surrogate for the classification error we consider the hinge loss $\ell(t,y) := \max\{0, 1-ty\}$, where $y \in \{-1, 1\}$, and we adopt the Gaussian kernel with scale parameter equal to 10 to cope with the nonlinearity of the problem, and let $r = 1$; we notice that this results in $L = 1$ and $\sigma^2 = 1$. We run Algorithm 4.1 with $w_k = \sqrt{k}$ for 100 iterations. The classifier's decision boundary is reported in Figure 3 while its estimated expected classification error, and a test set of 1000 examples, is about 0.1.

6. Conclusion. In this work we established high probability convergence rates for the projected stochastic subgradient method under heavy-tailed noise, that is, under the sole assumption that the stochastic subgradient oracle has uniformly bounded variance. We provide

a unified analysis which simultaneously cover general averaging schemes, stepsizes, and clipping levels, while avoiding large numerical constants in the statistical bounds and algorithm parameters. Moreover, we provided an application to the case of statistical learning with kernels which obtains near-optimal performances in a fully practicable fashion. Future interesting directions include avoiding the boundedness assumption on the constraint set and analyzing the last iterates, while keeping the simplicity and practicability of our parameter settings. Yet another valuable future research direction would be to extend our analysis to zero-order optimization settings.

REFERENCES

[1] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1999.

[2] A. Caponnetto and E. D. Vito, *Optimal rates for the regularized least-squares algorithm*, Found. Comput. Math., 7 (2007), pp. 331–368.

[3] A. Cutkosky, *Anytime online-to-batch, optimism and acceleration*, in Proceedings of the 36th International Conference on Machine Learning, 2019, pp. 1446–1454.

[4] D. Davis, D. Drusvyatskiy, L. Xiao, and J. Zhang, *From low probability to high confidence in stochastic convex optimization*, J. Mach. Learn. Res., 22 (2021), 49.

[5] A. Dieuleveut and F. Bach, *Nonparametric stochastic approximation with large step-sizes*, Ann. Statist., 44 (2016), pp. 1363–1399.

[6] Y. M. Ermol'ev, *On the method of generalized stochastic gradients and quasi-Féjer sequences*, Cybernetics, 5 (1969), pp. 208–220.

[7] D. A. Freedman, *On tail probabilities for martingales*, Ann. Probab., 3 (1975), pp. 100–118.

[8] S. Ghadimi and G. Lan, *Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization,* ii*: Shrinking procedures and optimal algorithms*, SIAM J. Optim., 23 (2013), pp. 2061–2089, https://doi.org/10.1137/110848876.

[9] E. Gorbunov, M. Danilova, and A. Gasnikov, *Stochastic optimization with heavy-tailed noise via accelerated gradient clipping*, in Proceedings of the 34th Annual Conference on Neural Information Processing Systems, 2020, pp. 15042–15053.

[10] E. Gorbunov, M. Danilova, I. Shibaev, P. Dvurechensky, and A. Gasnikov, *Near-optimal High Probability Complexity Bounds for Non-Smooth Stochastic Optimization with Heavy-Tailed Noise*, preprint, https://arxiv.org/abs/2106.05958, 2021.

[11] N. J. Harvey, C. Liaw, Y. Plan, and S. Randhawa, *Tight analyses for non-smooth stochastic gradient descent*, in Proceedinds of the 32nd International Conference on Computational Learning Theory, 2019, pp. 1579–1613.

[12] N. J. Harvey, C. Liaw, and S. Randhawa, *Simple and Optimal High-Probability Bounds for Strongly-Convex Stochastic Gradient Descent*, preprint, https://arxiv.org/abs/1909.00843, 2019.

[13] M. J. Holland, *Anytime guarantees under heavy-tailed data*, in Proceedings of the 36th AAAI Conference on Artificial Intelligence, 2022, pp. 6918–6925.

[14] P. Jain, D. Nagaraj, and P. Netrapalli, *Making the last iterate of SGD information theoretically optimal*, in Proceedinds of the 32nd International Conference on Computational Learning Theory, 2019, pp. 1752–1755.

[15] G. Lan, *First-order and Stochastic Optimization Methods for Machine Learning*, Vol. 1, Springer, 2020.

[16] J. Lin, R. Camoriano, and L. Rosasco, *Generalization properties and implicit regularization for multiple passes SGM*, in International Conference on Machine Learning, PMLR, 2016, pp. 2340–2348.

[17] J. Lin and L. Rosasco, *Optimal learning for multi-pass stochastic gradient methods*, Adv. Neural Inf. Process. Syst., 18 (2017), pp. 1–47.

[18] A. V. Nazin, A. S. Nemirovsky, A. B. Tsybakov, and A. B. Juditsky, *Algorithms of robust stochastic optimization based on mirror descent method*, Automat. Remote Control, 80 (2019), pp. 1607–1627.

[19] A. S. Nemirovskij and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*, Wiley-Interscience, 1983.

[20] B. T. Poljak, *Subgradient methods: A survey of Soviet research*, Nonsmooth Optim., 3 (1978), pp. 5–29.

[21] B. T. Polyak, *A general method for solving extremal problems*, Dokl. Akad. Nauk SSSR, 8 (1967), pp. 593–597.
[22] P. Richtárik, I. Sokolov, and I. Fatkhullin, *Ef21: A new, simpler, theoretically better, and practically faster error feedback*, in Proceedings of the 34th Annual Conference on Neural Information Processing Systems, 2021, pp. 4384–4396.
[23] O. Shamir and T. Zhang, *Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes*, in Proceedings of the 30th International Conference on Machine Learning, 2013, pp. 71–79.
[24] N. Z. Shor, *Minimization Methods for Non-differentiable Functions*, Springer, 1985.
[25] I. Steinwart and A. Christmann, *Support Vector Machines*, Springer, New York, 2008.
[26] I. Steinwart, D. R. Hush, and C. Scovel, *Optimal rates for regularized least squares regression*, in Proceedings of the 22nd Conference on Learning Theory, 2009, pp. 79–93.
[27] J. Zhang, S. P. Karimireddy, A. Veit, S. Kim, S. Reddi, S. Kumar, and S. Sra, *Why are adaptive methods good for attention models?*, in Proceedings of the 34th Annual Conference on Neural Information Processing Systems, 2020, pp. 15383–15393.