SIAM J. OPTIM.
Vol. 32, No. 2, pp. 795–821

DIFFERENTIALLY PRIVATE ACCELERATED OPTIMIZATION ALGORITHMS[*]

NURDAN KURU[†], Ş. İLKER BIRBIL[‡][¶], MERT GÜRBÜZBALABAN[§][¶], AND SINAN YILDIRIM[†][¶]

Abstract. We present two classes of differentially private optimization algorithms derived from the well-known accelerated first-order methods. The first algorithm is inspired by Polyak's heavy ball method and employs a smoothing approach to decrease the accumulated noise on the gradient steps required for differential privacy. The second class of algorithms are based on Nesterov's accelerated gradient method and its recent multistage variant. We propose a noise dividing mechanism for the iterations of Nesterov's method in order to improve the error behavior of the algorithm. The convergence rate analyses are provided for both the heavy ball and the Nesterov's accelerated gradient method with the help of the dynamical system analysis techniques. Finally, we conclude with our numerical experiments showing that the presented algorithms have advantages over the well-known differentially private algorithms.

Key words. differential privacy, accelerated optimization methods

AMS subject classifications. 68P27, 90C30, 90C25

DOI. 10.1137/20M1355847

1. Introduction.

In many real applications involving data analysis, the data owners and the data analyst may be different parties. In such cases, privacy of the data could be a major concern. Differential privacy promises securing an individual's data while still revealing useful information about a population [11]. It is based on constructing a mechanism for which output stays probabilistically similar whenever a new item is added or an existing one is removed from the dataset. Such incremental mechanisms have been shown to ensure data privacy [12]. Differential privacy is used within various types of methods in machine learning, such as boosting, linear and logistic regression, and support vector machines [15, 9, 36, 46].

In this work, we consider the scenario where a data analyst performs analysis on a dataset owned by another party by means of solving an optimization problem with (stochastic) first-order methods for empirical risk minimization. There is in fact a large body of work on differentially private empirical risk minimization [10, 24, 5, 47]. We will specifically focus on privacy-preserving gradient-based iterative algorithms, which are a popular choice for large-scale problems due to their scalability

[*]Received by the editors August 5, 2020; accepted for publication (in revised form) January 13, 2022; published electronically May 9, 2022. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.
https://doi.org/10.1137/20M1355847

Funding: The first author acknowledges support from a TÜBİTAK 2214A Scholarship which enabled an academic visit to Rutgers University. The third author's research was also supported in part by Office of Naval Research award N00014-21-1-2244 and National Science Foundation awards CCF-1814888 and DMS-2053485.

[†]Faculty of Engineering and Natural Sciences, Sabancı University, İstanbul, 34956, Turkey (nurdankuru@sabanciuniv.edu, sinanyildirim@sabanciuniv.edu).

[‡]Amsterdam Business School, University of Amsterdam, 1018 TV Amsterdam, The Netherlands (s.i.birbil@uva.nl).

[§]Department of Management Science and Information Systems, Rutgers University, Piscataway, NJ 08854 USA (mg1366@rutgers.edu).

[¶]These authors are listed in alphabetical order.

properties [1, 38, 41, 23, 42, 45]. Our contributions specifically regard two gradient-based stochastic accelerated algorithms, Polyak's heavy ball (HB) algorithm [33] and Nesterov's accelerated gradient (NAG) algorithm [30], as well as a recent variant of NAG [2].

Differential privacy can be achieved by adding carefully adjusted random noise to the input (data), such as in [19]; to the output (some function of data), such as in [9]; or to the iteration vectors of an iterative algorithm, such as in [1, 41, 23, 42, 32]. In this paper, we focus on the latter case in connection with gradient-based algorithms, where the iteration vectors of a gradient-based algorithm are revealed at the intermediate steps. This scenario is particularly relevant, for example, when some assessment should be done publicly on the convergence of the algorithms, or when the available data are shared among multiple users. For the latter case, see, e.g., [23], where the authors tackle distributed learning via empirical risk minimization while protecting the privacy of data holders, and present an approach based on gradient perturbation. Although the intrinsic randomness in a stochastic gradient descent (SGD) algorithm has been shown to provide some level of privacy in a recent study [22], the authors report high levels of privacy loss for most datasets. That is why most of the studies in the literature consider adding a suitable noise vector to the gradient at each step. However, this noise does harm the performance of the algorithm in such a way that it may even cause divergence. Therefore, the utility of a privacy-preserving algorithm is always a concern, as in our work.

There has been a great deal of work on improving the utility of gradient-based algorithms while preserving a given amount of the privacy "budget" (a mathematical definition of this budget is given in section 2). A well-known computational tool is, for example, subsampling, which is analyzed in a broader context in [5]. Norm clipping, that is, bounding the norm of the gradient according to a threshold, is also used to control the amount of noise; see, for instance, [1, 32, 39]. Analytical developments are also present: The authors of [1] focus on tracking higher moments of the privacy loss to obtain tighter estimates on the privacy loss. Other forms of differential privacy are also employed to conduct tighter analysis of the privacy loss [14, 7, 28, 47, 42]. An example regarding empirical risk minimization is [42] which considers the l_0-constrained sparse learning problem, with applications over linear and logistic regression, and exploits the zero-concentrated differential privacy [7] for tight utility analysis.

Contributions. In this paper, we contribute to the existing literature on privacy-preserving gradient-based algorithms by proposing, and providing a theoretical analysis of, differentially private versions of HB and NAG.

Our first algorithm is a variant of HB, which employs a smoothing approach with the help of the information from the previous iterations. We use this mechanism to improve the privacy level by taking the weighted average of the current and the previous noisy gradients. We give a convergence rate analysis using the dynamical system analysis techniques for optimization algorithms [26, 21, 16]. Although this kind of analysis exists for the deterministic HB method [21], to the best of our knowledge, the case with noisy gradients has not been considered in the literature, except in [8], where a special case of quadratic objectives is studied for a particular choice of the stepsize and momentum parameters (corresponding to the traditional choice of parameters in deterministic HB methods). By expanding on [8, Theorem 12], we give general results in terms of the error bounds for any selection of stepsize and momentum parameters.

The main motivation behind our error analysis is to shed light on the effect that the free parameters in the algorithm, such as the stepsize and momentum parameters and the number of iterations, have on the performance. In the typical stochastic optimization setting, the noise in the gradients is assumed to have a bounded variance which does not depend on the number of iterations; therefore the performance bounds obtained for the accuracy of momentum-based algorithms such as NAG or HB (measured in terms of expected suboptimality of the iterates) with constant parameters can improve monotonically as the number of iterations is increased (see, e.g., [25, 8, 2]). However, this is not necessarily the case in privacy-preserving versions of these algorithms. This is because each iteration causes some privacy loss and the amount of noise in the gradients has to be increased as the total number of iterations increases. Likewise, it is not clear how to set the stepsize and the momentum parameter for optimum performance of a privacy-preserving version of an algorithm because of the complex trade-off between the convergence rate and the additive error due to noise. We address such issues for the differentially private HB algorithm by providing performance bounds and error rates in terms of the number of iterations and the momentum parameters. We extend the existing results from the literature [21, 8] to provide an analysis for general stepsize and momentum parameter choices for both quadratic objectives as well as for smooth strongly convex objectives for the HB method under noisy gradients. In particular, tuning the stepsize and the momentum parameters to the level of desired privacy allows us to achieve better accuracy in the privacy setting compared to the traditional choice of parameters previously used for the deterministic HB method.

Our second contribution regards differentially private versions of NAG [30]. NAG can simply be made differentially private by merely adding noise to its gradient calculations. However, how to distribute the privacy-preserving noise to iterations to ensure optimal performance has not been concretely addressed in the literature. This question can be reformulated as *how to distribute a given, fixed, privacy budget to the iterations of the algorithm.* The relevance of this question is due to the fact that in each iteration a noisy gradient is revealed, causing privacy loss. We address this problem for the differentially private versions of NAG. In doing so, we exploit some explicit bounds in [2] on the expected error of those algorithms when they are used with noisy gradients. Our findings show that distributing the privacy budget to iterations uniformly, which corresponds to using the same variance for the privacy-preserving noise for all iterations, is not the optimal way in terms of accuracy. We formally substantiate this claim in our work.

We also consider a differentially private version of a recent variant of NAG, the multistage accelerated stochastic gradient (MASG), introduced in [2] to improve error behavior. The method is tailored to deal with noisy gradients in NAG, and hence is quite relevant to our setting in which noise is used to help with preserving privacy. However, the authors have not considered differential privacy while designing their algorithm. Techniques similar to NAG will be used for the error analysis of the differentially private version of MASG. Moreover, our novel scheme of optimally distributing the privacy budget to the iterations can also be applied to MASG in a similar manner.

We would like to mention the techniques for, and the scope of, the analysis of our proposed algorithms. By their nature, the proposed algorithms are stochastic, where the gradient vector is augmented with privacy-preserving noise at each iteration. There exist several studies that analyze the convergence of stochastic accelerated algorithms; for instance, see [27, 20, 35] for works related to stochastic HB, and [44, 43, 29] for a unified analysis of stochastic versions of gradient descent (GD), NAG, and HB methods. We adopt a dynamical system representation approach that is preferred for analyzing the first-order optimization algorithms [26, 16, 21, 3, 2, 29]. In this approach, the convergence rate is found with respect to the rate of decrease of a Lyapunov function of the system state of the dynamic system induced by the algorithm.

Finally, we remark that the given results are satisfied even when the noise that corrupts the gradient is *uncorrelated* with the state of the algorithm, provided that the noise variance can be bounded. The case of uncorrelated noise is evidently more general than the case of independent noise. In our setting, uncorrelatedness of the noise in the gradient is ensured by the noise being zero mean with a bounded variance conditioned on the state of the algorithm. Such characteristics of the gradient noise are quite relevant to differential privacy for two reasons: First, subsampling is a common technique used in privacy-preserving algorithms, and the error due to subsampling has zero mean and its variance is typically dependent on the current iterate of the algorithm. Second, the variance of the privacy-preserving noise is adjusted by a so-called sensitivity function of the state of the algorithm, which may be state dependent.

While most of the works for differentially private empirical risk minimization, including those mentioned so far, are based on the standard gradient descent algorithm, [41] considers other gradient-based algorithms (such as mirror descent and stochastic variance reduced gradient) and presents utility bounds for the solution of the empirical risk minimization problem under various assumptions related to the objective function. The idea is similar to our idea of benefiting from the accelerated methods. In addition to employing different algorithms, theoretical analysis, and targeting a different form of privacy (known as (ϵ, δ) differential privacy), our work also differs from [41] in terms of taking the optimal distribution of the privacy budget into account.

2. Preliminaries.

In this section, we present the preliminaries for the gradient-based optimization algorithms that we consider in the paper, followed by an introduction of differential privacy and its relation to the presented optimization algorithms.

2.1. Gradient-based optimization.

A vast variety of problems in machine learning can be written as unconstrained optimization problems of the form

$$\min_{x \in \mathbb{R}^d} F(x), \tag{2.1}$$

where $x \in \mathbb{R}^d$ is a parameter vector of dimension $d \geq 1$. This paper concerns a data-oriented optimization problem, where the objective function depends on a given dataset $Y = \{y_1, \ldots, y_n\} \subseteq \mathcal{Y}$. The objective function in (2.1) is a sum of functions that correspond to contributions of the individual data points $y_1, \ldots, y_n$ to the global objective. More specifically, we are interested in objective functions of the form

$$F(x) = \frac{1}{n} \sum_{i=1}^{n} f(x; y_i), \tag{2.2}$$

where $f(\cdot; y) : \mathbb{R}^d \mapsto \mathbb{R}$ for $y \in \mathcal{Y}$. These problems arise in empirical risk minimization in the context of supervised learning [40]. Note that one could write $F(x; Y)$ in order to emphasize the dependency of F on Y. However, for the sake of simplicity, we suppress Y in the notation. In this paper, we further restrict our attention to the set of strongly convex and smooth (that is, with a Lipschitz continuous gradient) functions; see Definition A.1 in Appendix A.

Gradient-based methods are arguably the most popular methods for the optimization problem in (2.1). We define the gradient vectors for the additive functions

$$\nabla f(x;y) = \left(\frac{\partial f(x;y)}{\partial x_1}, \dots, \frac{\partial f(x;y)}{\partial x_d}\right)^\top, \quad x \in \mathbb{R}, \quad y \in \mathcal{Y},$$

so that the (*full*) gradient $\nabla F(x)$ is given by

$$\nabla F(x) = \frac{1}{n}\sum_{i=1}^{n} \nabla f(x; y_i). \tag{2.3}$$

The iterates of the basic gradient descent method for the solution of (2.1) are given by

$$x_{t+1} = x_t - \alpha \nabla F(x_t), \quad t \geq 0, \tag{2.4}$$

where α is the (constant) learning rate. There are two well-known modifications of the basic gradient descent method: Polyak's heavy ball (HB) method [33] and Nesterov's accelerated gradient (NAG) method [30]. Both introduce a momentum parameter $\beta \geq 0$ to improve upon the convergence of gradient descent. The update rule for HB at iteration t is given by

$$x_{t+1} = x_t - \alpha \nabla F(x_t) + \beta(x_t - x_{t-1}), \tag{2.5}$$

whereas the update rule for NAG at iteration t is simply

$$\begin{aligned} x_{t+1} &= z_t - \alpha \nabla F(z_t), \\ z_t &= (1+\beta)x_t - \beta x_{t-1}. \end{aligned} \tag{2.6}$$

There exist stochastic versions of these gradient-based methods that are employed when either the gradients are noisy or an exact calculation per iteration is too expensive. In the former case, $\nabla F(x_t)$ is simply replaced by the noisy gradient, provided that the noisy gradient is an unbiased estimator of the true gradient. In the second case, the computationally costly $\nabla F(x_t)$ is replaced by a minibatch estimator

$$\nabla F_{B_t}(x) := \frac{1}{m}\sum_{i \in B_t} \nabla f(x; y_i), \tag{2.7}$$

where B_t is a subset $B \subseteq \{1, \dots, n\}$ with $|B_t| = m$, formed by sampling without replacement so that $\nabla F_{B_t}(x)$ is unbiased.

2.2. Differential privacy. In our subsequent discussion, we will modify the steps of the gradient-based methods to have privacy-preserving updates. Our setting is as follows: The data holder makes public the iterates $\{x_t\}_{0 \leq t \leq T}$ for a total of T iterations. The algorithm is known with all its parameters α (and β). If the data holder applies the related update of the method directly, the vectors $\nabla F(x_t)$ are revealed. This violates privacy since the revealed terms are deterministic functions of the data. Therefore, due to privacy concerns, the iterates have to be randomized by using a noisy gradient.

Differential privacy quantifies the privacy level that one guarantees by such randomizations. A randomized algorithm takes an input dataset $Y \in \mathcal{Y}$ and returns the random output $A_Y \in \mathcal{X}$. Such an algorithm can be associated with a function

$\mathcal{A}: \mathcal{Y} \to \mathcal{P}$ that maps a dataset from $\mathcal{Y}$ to a probability distribution $\mathcal{A}(Y) \in \mathcal{P}$ such that the output is random with $A_Y \sim \mathcal{A}(Y)$. For datasets Y_1 and Y_2, let $h(Y_1, Y_2)$ denote the Hamming distance between Y_1 and Y_2. This distance indicates the number of different elements between the two datasets. A differentially private algorithm ensures that $\mathcal{A}(Y_1)$ and $\mathcal{A}(Y_2)$ are "not much different" if $h(Y_1, Y_2) = 1$. This statement is formally expressed by [12] (see Definition A.2 in Appendix A).

Most existing differentially private methods perturb certain functions of data with a suitably chosen random noise. The amount of this noise is related to the sensitivity of the function, which is the maximum amount of change in the function when one single entity of the data is changed (see Definition A.3 in Appendix A). There are many results proposed in the literature that provide differential privacy for iterative algorithms. Among those results, we will mainly use three of them concerning Laplace mechanism, composition, and subsampling. For ease of reference, the corresponding three theorems are also given in Appendix A.

When privacy is of concern for the optimization problem (2.1), one approach is to update the parameter x_t of iteration t using a noisy (stochastic) gradient vector

$$\widetilde{\nabla F_{B_t}}(x_t) = \nabla F_{B_t}(x_t) + \eta_t, \tag{2.8}$$

where B_t are the indices of full (sampled) data with size m, and $\eta_t = (\eta_{t,1}, \dots, \eta_{t,d})^\top$ is a vector of independent noise terms having Laplace distribution with its parameter value chosen suitably to provide the desired level of privacy. Although the privacy of an algorithm can be guaranteed in this way, the performance will be affected because of the noise added at each iteration. In this paper, we analyze the present trade-offs between accuracy and privacy in gradient-based algorithms and propose accelerated algorithms with good performance under the differential privacy noise.

Privacy setting. Before proceeding to the algorithms, we concretely state the privacy setting assumed throughout the paper. This setting has been adopted in many works; see, e.g., [1, 32, 39]. When an algorithm is run for a total of T iterations, the entire sequence of the iterates outputted by the algorithm, $x_0, x_1, \dots, x_T$, is made available to the *adversary*, who is any hypothetical entity with unbounded computational capability who wants to violate the privacy of the sensitive data. Moreover, the algorithm itself is known with all its parameters (such as α, β, and the variance of the noise added to the gradient). Therefore, the adversary's view is the sequence of iterates as well as the algorithm details. The dataset $y_{1:n}$ is sensitive, and it is unavailable to the adversary. We also assume that $y_{1:n}$ is constructed with contributions of personal (private) data from n individuals, where y_i is the ith individual's data.

It is worth noting here that revealing $x_1, \dots, x_T$ is equivalent to revealing the noisy gradients $\widetilde{\nabla F_{B_0}}(x_0), \dots, \widetilde{\nabla F_{B_{T-1}}}(x_{T-1})$ in (2.8), since one sequence can be constructed from the other given the initial point x_0 and the algorithm details. That is why in the subsequent privacy analysis we will focus on the noisy gradients which are easier to handle in deriving the privacy properties of the algorithms.

3. Differentially private heavy ball algorithm.

We start with investigating a differentially private version of the stochastic HB algorithm, which we will abbreviate as DP-SHB. The update rule of this algorithm operates on a dataset of size n with steps

$$x_{t+1} = x_t - \alpha(\nabla F_{B_t}(x_t) + \eta_t) + \beta(x_t - x_{t-1}), \tag{3.1}$$

where $0 < \beta < 1$ is the momentum parameter of HB, B_t's are i.i.d. random subsamples of size $m \le n$ sampled without replacement, and η_t's are independent random vectors having i.i.d. noise components with Laplace$(b_t(x_t))$, which is the Laplace distribution with a zero mean and variance $2b_t(x_t)^2$. Here, differential privacy of (3.1) is sought through the noisy gradient $\nabla F_{B_t}(x_t) + \eta_t$. The minimum value, required for the parameter $b_t(x_t)$ of the Laplace distribution to have ϵ-differential privacy, depends on the number of iterations T, the subsample size m, and the L_1 sensitivity $S_1(x_t)$ at x_t, where the L_1 sensitivity function is defined as

$$S_1(x) = \sup_{y,y' \in \mathcal{Y}} |\nabla f(x;y) - \nabla f(x;y')|, \quad x \in \mathbb{R}^d. \tag{3.2}$$

Observing (2.7), we see that changing Y and Y' in one data item corresponds to the existence of a single pair of different values (y_i, y_i'). Hence, the change in $F(x)$ is by at most $S_1(x)/n$.

Consider the DP-SHB algorithm, where at iteration t, we draw a subsample of size m from a dataset of size n and add Laplacian noise with parameter $b_t(x_t)$ to the minibatch estimator in (2.8). Then, using the result regarding the Laplace mechanism in Theorem A.1 and the privacy amplification result stated in Theorem A.3, the privacy loss at the iteration can be shown to be

$$\epsilon_t = \varepsilon(S_1(x_t), b_t(x_t), n, m),$$

where the function $\varepsilon : [0,\infty)^2 \times \{(m,n) \in \mathbb{Z}_+ : m \leq n\} \mapsto \mathbb{R}$ is given as

$$\varepsilon(S,b,n,m) := \ln\left[(e^{S/(bm)} - 1)\frac{m}{n} + 1\right] \quad \text{for} \quad S, b \in [0,\infty)^2; m \leq n \in \mathbb{Z}_+. \tag{3.3}$$

Note that, for $m = n$, i.e., under no subsampling, we end up with $\varepsilon(S,b,n,n) = S/(bn)$. The following proposition uses this fact and states the required amount of noise variance in order to have an ϵ-differentially private algorithm after T iterations.

PROPOSITION 3.1. *The DP-SHB algorithm in* (3.1) *leads to an ϵ-differentially private algorithm if the parameter $b_t(x_t)$ of the Laplace distribution* Laplace$(b_t(x_t))$ *for each component of the noise vector η_t at iteration t is chosen as*

$$b_t(x_t) = \frac{S_1(x_t)}{m\epsilon_0}, \tag{3.4}$$

where x_t is the output value at iteration t, n is the number of data points,

$$\epsilon_0 = \ln\left[1 + (e^{\epsilon/T} - 1)n/m\right], \tag{3.5}$$

m is the subsample size, and T is the maximum number of iterations.

Proof. Using the $b_t(x_t)$ given in the proposition, the privacy loss in one iteration is $\varepsilon(S_1(x), b_t(x_t), n, m) = \ln\left[1 + (m/n)\left(e^{\epsilon_0} - 1\right)\right] = \epsilon/T$. Finally, we apply Theorem A.2 to conclude that the privacy loss after T iterations is ϵ. □

We are interested in DP-SHB because it lends itself to an interpretation quite relevant to the differential privacy setting. The noise used in the differentially private versions of the gradient descent algorithm has to be higher as the number of iterations grows, i.e., $b_t(x_t)$ needs to be larger for a larger T. This can be seen from (3.4). One way to reduce the required noise is to use a smoothed noisy gradient, where the smoothing is recursively performed on the past and current gradient estimates. This is indeed how DP-SHB works. The update in (3.1) can be rewritten as

$$x_{t+1} = x_t - \frac{\alpha}{1-\beta}\bar{u}_t, \tag{3.6}$$

where $\bar{u}_t$ is a geometrically weighted average of all the gradients up to the current iteration defined recursively as

$$\bar{u}_t = \beta \bar{u}_{t-1} + (1-\beta)\left(\nabla F_{B_t}(x_t) + \eta_t\right) \tag{3.7}$$

with the initial condition $\bar{u}_0 = (1-\beta)(\nabla F_{B_0}(x_0) + \eta_0)$. We note that a smoothing strategy similar to that in DP-SHB, which combines minibatching with a noise-adding mechanism for averaged gradients, has been used in [31], albeit in a different setting, namely for the purpose of private variational Bayesian inference.

3.1. Analysis of DP-SHB. For analyzing the convergence of DP-SHB, we first cast it as a dynamical system. We introduce the (random) variable

$$v_t = \nabla F(x_t) - \nabla F_{B_t}(x_t),$$

which accounts for the error due to subsampling. Using this definition, we can write

$$x_{t+1} = x_t - \alpha(\nabla F(x_t) + \eta_t + v_t) + \beta(x_t - x_{t-1}). \tag{3.8}$$

Then, the dynamical system representation of DP-SHB becomes

$$\begin{aligned} \xi_{t+1} &= [A \otimes I_d]\xi_t + [B \otimes I_d](u_t + v_t + \eta_t), \\ z_t &= [C \otimes I_d]\xi_t, \\ u_t &= \nabla F(z_t), \end{aligned} \tag{3.9}$$

where I_d is the $d \times d$ identity matrix, $\otimes$ denotes the Kronecker product, and the state vector ξ_t and the system matrices A, B, and C are given as

$$\xi_t = \begin{bmatrix} x_t \\ x_{t-1} \end{bmatrix}, \quad A = \begin{bmatrix} 1+\beta & -\beta \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{3.10}$$

In our error analysis, we will consider both stochastic and deterministic versions of HB. In order to do that, we need a uniform bound (in x_t) for the conditional covariance of $w_t := \eta_t + v_t$ given x_t (for the case without subsampling, we simply take $v_t = 0$). Note that, due to independence of η_t and v_t conditional on x_t, the conditional covariance of w_t given x_t satisfies

$$\mathrm{Cov}(w_t|x_t) = \mathrm{Cov}(\eta_t|x_t) + \mathrm{Cov}(v_t|x_t).$$

To handle the contribution to the overall noise by the privacy-preserving noise η_t, we make the following assumption.

Assumption 3.2 (bounded L_1 sensitivity). *The L_1 sensitivity function defined in* (3.2) *is bounded in x. That is, there exists a scalar constant S_1 such that*

$$\sup_{x \in \mathbb{R}^d} S_1(x) \leq S_1. \tag{3.11}$$

Assumption 3.2 is common in the differential privacy literature that specifies an upper bound for error analysis. We note that one can give a particular bound using the properties of the objective (loss) function. For example, the logistic regression model, which we will use to show our numerical experiments in section 5, easily admits such a bound. It turns out that Assumption 3.2 readily guarantees a bound on the variance of v_t, the subsampling noise. The next proposition formally shows this observation. The proof is given in Appendix B.1.

PROPOSITION 3.3. *If Assumption* 3.2 *holds, the norm of the conditional covariance of $w_t = \eta_t + v_t$ is bounded for all t uniformly in x_t as*

$$||\mathrm{Cov}(w_t|x_t)|| \leq \mathcal{E}_T := \sigma_s^2(m,n) + 2\frac{dS_1^2}{m^2\epsilon_0^2}, \tag{3.12}$$

where ϵ_0 is given in (3.4) *and $\sigma_s^2(m,n)$ is an upper bound on the norm of the covariance of the error due to subsampling given by*

$$\sigma_s^2(m,n) = \frac{S_1^2}{4}\frac{1}{m}\frac{n-m}{n-1}. \tag{3.13}$$

Note that $\mathcal{E}_T$ depends on the total number of iterations T through ϵ_0, hence the subscript.

Before going into the detailed technical analysis, we find it useful to provide a sketch of it. Our purpose is to find an upper bound for the expected suboptimality $\mathbb{E}[F(x_t) - F^*]$, where x^* is the optimal solution of (2.1) and $F^* := F(x^*)$ is the minimum value of F. The upper bound we will prove is of the form

$$\mathbb{E}[F(x_t) - F^*] \leq \rho^{2t}\psi_0 + \mathcal{E}_T R, \quad 0 \leq t \leq T,$$

for some rate ρ, a nonnegative ψ_0 that is related to the initial point x_0, and a nonnegative R. As we will show soon, this bound in the DP setting has interesting aspects: Note that, as an issue unique to the differential privacy context, *the term $\mathcal{E}_T$ increases with the total number of iterations, T.* This is because for fixed privacy level ϵ, as T increases, ϵ_0 defined in (3.4) decreases. Hence, increasing the number of iterations T makes the first term $\rho^{2T}\psi_0$ smaller; however it leads to an increase in the second term $\mathcal{E}_T R$. This makes the analysis of DP-SHB fundamentally different compared to the analysis of the standard SHB in the stochastic optimization literature (see, e.g., [8, 20, 18]), where the second term is scaled with the fixed noise variance parameter that does not change with the number of iterations.

For analysis purposes, we define $\bar{F} : \mathbb{R}^{2d} \mapsto \mathbb{R}$ such that for $\xi_t = \begin{bmatrix} x_t^\top & x_{t-1}^\top \end{bmatrix}^\top$, we have $\bar{F}(\xi_t) = F(x_t)$. Also, for a 2×2 symmetric positive-definite matrix P and a positive scalar c, we set the Lyapunov function

$$V_{P,c}(\xi) = V_P(\xi) + c(\bar{F}(\xi) - F^*)$$

with $V_P(\xi) = (\xi - \xi^*)^\top [P \otimes I_d](\xi - \xi^*)$. The following proposition, which is constructed in a similar vein as Proposition 4.6 in [3], allows us to obtain expected suboptimality bounds depending on the parameters α and β as well as the noise level $\mathcal{E}_T$ and a convergence rate ρ. A proof is given in Appendix B.2.

PROPOSITION 3.4. *Given $F \in \mathcal{S}_{\mu,L}(\mathbb{R}^d)$, consider running the DP-SHB algorithm with constant parameters α and β for T iterations and with $b_t(x_t)$ in Proposition* 3.1 *so that ϵ-differential privacy is satisfied. Suppose that Assumption* 3.2 *holds and there exist $\rho \in (0,1)$, a 2×2 positive-semidefinite symmetric matrix P, and constants $c_0, c \geq 0$ such that*

$$c_0 X_0 + c[X_1 + (1-\rho^2)X_2] \succeq \Phi(A,B,P,\rho), \tag{3.14}$$

where

$$X_0 = \begin{bmatrix} 2\mu L C^\top C & -(\mu+L)C^\top \\ -(\mu+L)C & 2I_d \end{bmatrix}, \quad \Phi(A,B,P,\rho) = \begin{bmatrix} A^\top PA - \rho^2 P & A^\top PB \\ B^\top PA & B^\top PB \end{bmatrix},$$

the matrices A, B, C are as defined in (3.10)*, and*

$$X_1 = \frac{1}{2}\begin{bmatrix} -L\beta^2 & L\beta^2 & -(1-L\alpha)\beta \\ L\beta^2 & -L\beta^2 & (1-L\alpha)\beta \\ -(1-L\alpha)\beta & (1-L\alpha)\beta & \alpha(2-L\alpha) \end{bmatrix}, \quad X_2 = \frac{1}{2}\begin{bmatrix} \mu & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}.$$

Then, for all $0 \leq t \leq T$, we obtain

$$\mathbb{E}[F(x_t) - F^*] \leq \rho^{2t}\frac{1}{c}V_{P,c}(\xi_0) + \frac{1-\rho^{2t}}{1-\rho^2}\frac{Ld\alpha^2}{2}\mathcal{E}_T\left(1 + \frac{2P_{12}^2}{P_{22}cL + 2|P|}\right), \tag{3.15}$$

where $\mathcal{E}_T$ is defined as in (3.12)*, $|P|$ denotes the determinant of P, and we have the convention $0/0 = 0$ for the last factor.*

As distinct from the approach in [3], which is developed for the NAG method, the bound in (3.15) is constructed by adapting the results for the deterministic HB [21] to the stochastic setting. We also note that the matrix inequality (3.14) is 3×3 and can be solved numerically for ρ and P in practice by a simple grid search over the rate ρ and entries of the 2×2 matrix P (see, e.g., [21, 26, 8]). Therefore, the right-hand side of (3.15) that provides performance bounds can be computed numerically in practice.

3.2. Analysis of quadratic objective function case. In this section, we will present explicit bounds for a quadratic objective function in order to provide more insight into the interplay among α, β, and the number of iterations T. We consider the following quadratic function:

$$F(x) = \frac{1}{2}x^\top Qx + a^\top x + b, \tag{3.16}$$

where $Q \in \mathbb{R}^{d\times d}$ is symmetric positive-definite, $a \in \mathbb{R}^d$ is a column vector, and $b \in \mathbb{R}$ is a scalar. For such a strongly convex quadratic objective function, an exact bound for the objective error can be presented.

To put it in a differential privacy context, we can assume that the parameters of $F(x)$ depend on some data $Y = \{y_1, \dots, y_n\}$. For example, F is a sum of functions $f(\cdot; y_i)$ that are quadratic in x (hence F itself is quadratic in x), and the coefficients of the quadratic expression for each $f(\cdot; y_i)$ depend on y_i. We will assume that the L_1 sensitivity of F is such that the required DP noise satisfies $\mathbb{E}(\eta_t\eta_t^\top) = \sigma_T^2 I_d$ for some $\sigma_T^2 > 0$. For simplicity, we assume that no subsampling is performed, i.e., $v_t = 0$.

The optimal values for HB in the nonnoisy setting have been given in [34] as $\alpha_{\text{HB}} = 4/(\sqrt{\mu} + \sqrt{L})^2$ and $\beta_{\text{HB}} = (\sqrt{\kappa} - 1)^2/(\sqrt{\kappa} + 1)^2$ where $\kappa := L/\mu$. However, those "optimal" values may not be the best selection for α and β for DP-SHB. There are two reasons for this. First, due to privacy concerns, noise is inevitable in DP-SHB. Presence of noise shows as a second additive term in the bound for the error. This second term is affected by the selection of α. Second, the amount of privacy-preserving noise increases with the total number of iterations. In general, the error bound is a sum of two terms. The first of these decreases with the convergence rate ρ of the algorithm, and the second term is due to privacy-preserving noise. It will be shown that α and β have an influence on both the convergence rate and the multiplicative constant of the additive error due to noise. We will additionally see that a selection of the α, β pair that improves the rate also increases the additive error term due to the presence of privacy-preserving noise. Therefore, we can talk about a trade-off between the convergence rate and the additive noise term in our performance bounds, which is adjusted by the parameters α and β. In that respect, the "optimal" choice of α and β in the nonnoisy setting is typically not the best choice of α and β in the DP setting.

By adapting [8, Theorem 12], which was given for the parameter choices $\alpha_{\text{HB}}, \beta_{\text{HB}}$, we present our result for the error bound given by any pair α, β. A proof is given in Appendix B.3.

THEOREM 3.5. *Let $F \in \mathcal{S}_{\mu,L}(\mathbb{R}^d)$ be a quadratic function given in* (3.16)*. Consider the iterates $\{x_t\}_{0\leq t\leq T}$ of the DP-SHB method, which is run for T iterations with noisy gradients $\nabla F(x_t) + w_t$ where $\mathbb{E}(w_t|x_t) = 0$ and $\mathbb{E}(w_t w_t^\top|x_t) \preceq \sigma_T^2 I$ for some positive constant $\sigma_T^2 > 0$. If DP-SHB is run with parameters (α, β), then*

$$\mathbb{E}[F(x_t)] - F(x^*) \leq V(\xi_0)C_t^2\rho^{2t} + Lm(\alpha, \beta), \tag{3.17}$$

where

$$m(\alpha, \beta) = \frac{\sigma_T^2}{2}\sum_{i=1}^{d}\frac{2\alpha(1+\beta)}{(1-\beta)\lambda_i(2+2\beta-\alpha\lambda_i)}$$

with λ_i's being the eigenvalues of Q. In (3.17)*, we have*

$$\rho = \max\{|a_{\mu,+}|, |a_{\mu,-}|, |a_{L,+}|, |a_{L,-}|\}, \tag{3.18}$$

where

$$a_{\lambda,\pm} = \frac{1+\beta-\alpha\lambda \pm \sqrt{(1+\beta-\alpha\lambda)^2 - 4\beta}}{2},$$

and $V(\xi_0)$ is given by

$$V(\xi_0) = \mathbb{E}[\|(\xi_0 - \xi^*)(\xi_0 - \xi^*)^\top\|] + \frac{\sigma_T^2\alpha^2}{1-\rho^2}$$

with $C_t = \mathcal{O}(t)$ being a sequence of scalar coefficients, provided that $\rho < 1$.

Theorem 3.5 is a general result which holds for SHB for (3.16) with any choice of σ_T^2. The relation between Theorem 3.5 and differential privacy lies in the way σ_T^2 is determined in order to satisfy differential privacy with a given ϵ. More concretely, σ_T^2 depends on ϵ as well as T, sensitivity S_1, and the subsample size m. For example, if $S_1 = 1$ and $m = n$, then we need $\sigma_T^2 = (T/n\epsilon)^2$ to provide ϵ-differential privacy.

Note that in Theorem 3.5 we considered the case with uncorrelated and bounded noise variance, which generalizes over the independent noise setting. To the best of our knowledge, such a result has not been shown before in the literature.

Numerical demonstration. Here, we illustrate the effect of algorithm parameters over the error bound given in Theorem 3.5. The dimension of the objective function is taken as $d = 2$, and Q is chosen as the 2×2 diagonal matrix with $\mu = 0.5$ and $L = 1$ on its diagonal, so that its eigenvalues are μ and L.

We take $C_t = t$ for simplicity of the presentation.[1] With fixed stepsizes $\alpha \leq 1/L$, the convergence rate ρ in (3.18) versus β is plotted in Figure 1 for several values of α. As for the noise variance, we considered $\sigma_T^2 = (Tc_w)^2$ to represent increasing noise variance in the total number of iterations. We repeated our experiments for two different values of c_w, namely for $c_w = 10^{-4}$, representing a less noisy, hence less private, scenario (larger ϵ), and for $c_w = 10^{-2}$, representing a more noisy, hence more private, scenario (smaller ϵ). We observe that the "optimal" β value in terms of convergence rate ρ (which is indicated at the bottom row of Figure 1) shows a reliable performance.

[1] C_t is a constant multiple of t, but the constant in front of t would not change the qualitative behavior of the plots, only shifting the graphs by a constant factor in the logarithmic scale.

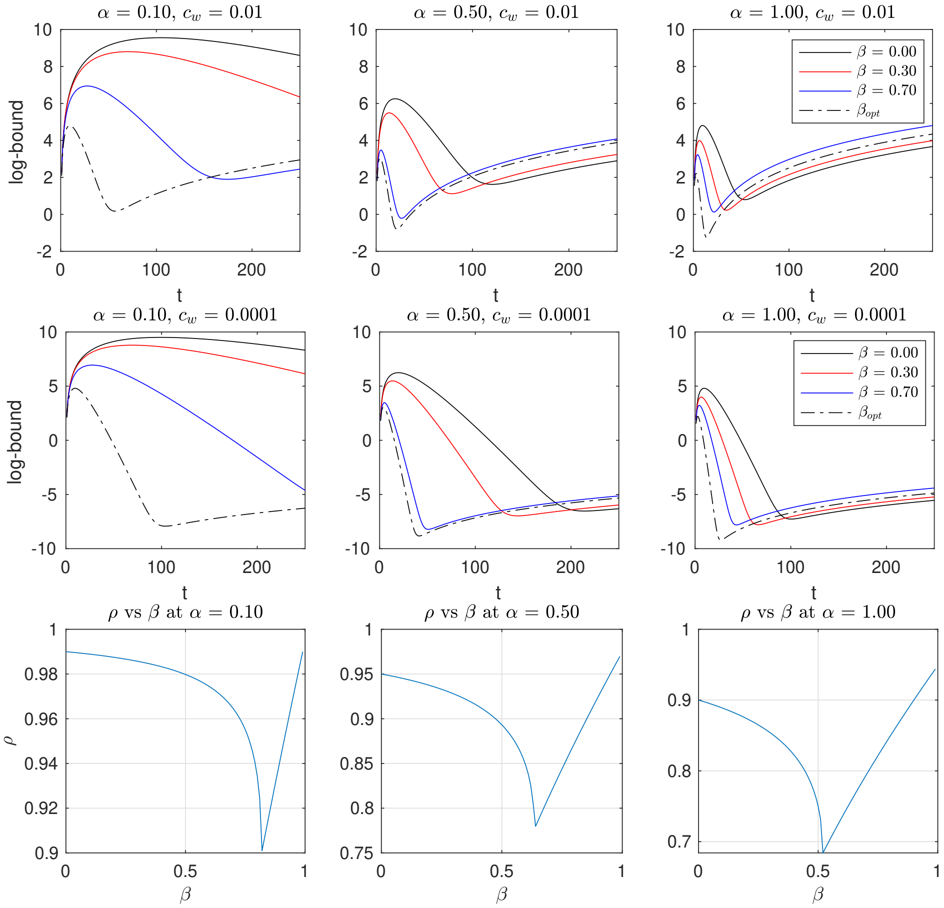

FIG. 1. *DP-SHB performance for the quadratic objective function case.*

4. Differentially private accelerated algorithms. In this section, we will investigate NAG in a differential privacy setting and propose two methods to tailor it for improved performance under differential privacy. The first method, presented in section 4.1, concerns how to distribute the privacy budget on the iterations to have the best results in terms of accuracy. Specifically, instead of distributing the privacy budget on the iterations evenly, we show in Proposition 4.2 how to allocate a given privacy budget on the iterations to optimize accuracy. The second method, presented in section 4.2, concerns varying the stepsize and momentum parameters of the NAG with iterations.

In the following discussion, we will assume that Assumption 3.2 on the existence of an upper bound S_1 on the sensitivity holds, as in the previous section. Furthermore, we will assume that the upper bound S_1 is considered while determining the parameter b_t of the privacy-preserving Laplace, so that b_t is independent from the current state. Using a state-independent sensitivity to determine the Laplace parameter is not uncommon, especially when it is hard to identify $S_1(x)$ for all x. An example illustrating this case can be found in section 5, in particular the sensitivity bound in (5.2) for the logistic regression model, which is independent of the state x.

Recall the NAG update in (2.6). A straightforward differentially private version of NAG would be obtained by cluttering the gradient with the privacy-preserving noise, just as in the DP-SHB algorithm. The corresponding change in NAG would be

$$\begin{aligned} x_{t+1} &= z_t - \alpha(\nabla F(z_t) + \eta_t), \\ z_t &= (1+\beta)x_t - \beta x_{t-1}, \end{aligned} \tag{4.1}$$

where $\eta_{t,i} \overset{\text{i.i.d.}}{\sim} \text{Laplace}(b)$ for $i = 1, \dots, d$, and $b = \frac{S_1}{n\epsilon_0}$ with ϵ_0 is given in (3.5). The resulting algorithm will be referred to as DP-NAG.

In DP-NAG, the stepsize α (hence β) and the DP noise parameter b are taken as constant. That begs the question of whether the performance of DP-NAG could be improved if we let b and α depend on t, the iteration number. We propose two methods to improve the performance of DP-NAG while preserving the same level of privacy. The first method seeks to improve the algorithm by making the DP variance parameter b, hence the privacy loss per iteration, dependent on the iteration number. See Proposition 4.2 for an explicit result for the no-subsampling case ($m = n$), which, interestingly, suggests in particular that distributing the given privacy budget on the iterations evenly is not the best way. The second method considers varying α (hence β) with iterations.

4.1. NAG with optimized DP variance.

We first present an error bound for NAG that uses noisy gradients. Let $E_t = \mathbb{E}(F(x_t)) - F^*$. The following theorem is adapted from [2, Theorem 2.3].

THEOREM 4.1. *Let $F \in \mathcal{S}_{\mu,L}(\mathbb{R}^d)$, and suppose that Assumption 3.2 holds. Consider a stochastic version of the NAG algorithm that runs with a stepsize $\alpha \le 1/L$ and the momentum parameter $\beta = (1-\sqrt{\alpha\mu})/(1+\sqrt{\alpha\mu})$ and uses noisy gradients $\widetilde{\nabla F(z_t)} = \nabla F_{B_t}(z_t) + \eta_t$ for $t \ge 0$ as in (2.8) with a subsampling size $m \le n$ and $\eta_{t,i} \overset{\text{i.i.d.}}{\sim} \text{Laplace}(b_t)$ for all $i = 1, \dots, d$. Then, for any $t \ge 1$, we have*

$$E_t \le (1-\sqrt{\mu\alpha})E_{t-1} + \alpha(1+\alpha L)\left(b_t^2 d + \sigma_s^2(m,n)/2\right). \tag{4.2}$$

Note that in (4.2), the term $b_t^2 + \frac{S_1^2}{m}\frac{n-m}{n-1}$ is an upper bound on the norm of the covariance of the gradient estimator, and it simplifies to b_t^2 when $m = n$, i.e., without subsampling. By starting the recursion in (4.2) at the last iteration $t = T$ and recursing backward until $t = 0$, we end up with

$$E_T \le (1-\sqrt{\mu\alpha})^T E_0 + \sum_{t=1}^{T}(1-\sqrt{\mu\alpha})^{T-t}\alpha(1+\alpha L)\left(b_t^2 d + \sigma_s^2(m,n)/2\right).$$

It will prove useful later to express the error of NAG generically as

$$E_T \le a_{T,0}E_0 + \sum_{t=1}^{T} a_{T,t}\left(b_t^2 d + \sigma_s^2(m,n)/2\right). \tag{4.3}$$

The $a_{T,t}$ in (4.3) can be identified as

$$a_{T,t} = \begin{cases} (1-\sqrt{\mu\alpha})^T, & t = 0, \\ (1-\sqrt{\mu\alpha})^{T-t}\alpha(1+\alpha L), & t = 1, \dots, T. \end{cases}$$

In the DP framework, we have control on the noise parameters b_t, with a constraint due to our privacy budget ϵ. Suppose that we are committed to running the algorithm for a total of T iterations. When b_t is used, the privacy loss at iteration t becomes $\epsilon_t = \varepsilon(S_1, b_t, n, m)$. Given a desired privacy level ϵ, we have the constraint $\sum_{t=1}^{T} \epsilon_t = \epsilon$, by Theorem A.2. Therefore, one question is, with fixed m and T, how we should arrange b_t so that the bound in (4.3) is optimized. Factoring our privacy budget into the scene, we have the following constrained optimization problem:

$$\min_{b_1,\dots,b_T} \sum_{t=1}^{T} a_{T,t} b_t^2, \quad \text{subject to } \sum_{t=1}^{T} \varepsilon(S_1, b_t, n, m) = \epsilon. \tag{4.4}$$

For general $m \neq n$, the constrained optimization problem is analytically intractable and needs a numerical solution. This is due to the nonlinearity in $\varepsilon(S_1, b_t, n, m)$. However, for the special case of $m = n$ (no subsampling), the constraint in (4.4) simplifies to $\sum_{t=1}^{T} S_1/nb_t = \epsilon$, allowing for the following tractable result. (A proof is given in Appendix B.4.)

PROPOSITION 4.2. *When $m = n$, the optimization problem in* (4.4) *is solved by*

$$b_t = \frac{\sum_{j=1}^{T} a_{T,j}^{1/3}}{a_{T,t}^{1/3}} \frac{S_1}{n\epsilon}, \quad t = 1, \dots, T. \tag{4.5}$$

Note that the choice of the b_t values is directly related to how we distribute the privacy budget over the iterations. Indeed, we can express the solution (4.5) also in terms of the privacy loss at iteration t as

$$\epsilon_t = \frac{a_{T,t}^{1/3}}{\sum_{j=1}^{T} a_{T,j}^{1/3}} \epsilon, \quad t = 1, \dots, T.$$

Since $a_{T,t}$ is decreasing in t, the solution (4.5) suggests that the variance should start high and then should be decreased. This means that the privacy budget should be distributed to the iterations in an uneven way. A larger part of the privacy budget should be spent for later rather than for early iterations.

Remark 4.1. The solution in (4.5) for $m = n$ yields the optimum bound

$$E_T \leq a_{T,0} E_0 + \frac{d S_1^2}{n^2 \epsilon^2} \left(\sum_{j=1}^{T} a_{T,j}^{1/3} \right)^3, \tag{4.6}$$

which is the optimized version of the bound in (4.3) with respect to the b_t values subject to the constraint in (4.4). This optimal bound could further be optimized with respect to the number of iterations, provided that one has an accurate guess on the initial error E_0. We note that increasing the number of iterations may degrade the performance in the DP context, since the required noise per iteration increases, unlike in the deterministic setting where one may improve the performance monotonically as the number of iterations grows.

Remark 4.2. Although the result in Proposition 4.2 is valid for no subsampling, it can be used as a guide for arranging b_t's even under subsampling. Note that for values of m, n, S, and b such that $m \ll n$ and $S/bm \ll 1$, we have $\varepsilon(S, b, m, n) \approx S/bn$, owing to the approximation $e^z \approx 1 + z$ for $z \ll 1$.

4.2. Multistage NAG. An alternative for improving the performance of NAG is to make the stepsize vary with iterations. In fact, the MASG algorithm of [2] has been proposed with that motivation. The authors prove that MASG achieves the optimal rate in both deterministic and stochastic versions.

In this paper, we present a DP-MASG, a differentially private version of MASG introduced in [2]. In order to study and improve the error behavior of the algorithm, an explicit bound for the objective error that accommodates iteration-dependent noise variance parameter b_t is presented. We demonstrate that the approach of dividing noise into iterations can be applied to MASG as well.

The original algorithm MASG is a multistage accelerated algorithm which uses the NAG method with noisy full gradient. The total iterations T are divided into K stages, with stage lengths n_k, and for each stage a different stepsize $\alpha^{(k)}$ is used. For the optimal convergence rate, the stage lengths and the corresponding stepsizes are recommended in [2] as

$$n_1 \geq 1, \quad \alpha^{(1)} = \frac{1}{L}, \quad n_k = 2^k \left\lceil \sqrt{\kappa} \ln(2^{p+2}) \right\rceil, \quad \alpha^{(k)} = \frac{1}{2^{2k} L}, \quad k \geq 2, \tag{4.7}$$

where $p \geq 1$.

The MASG algorithm can easily be modified to be differentially private by adding a Laplace noise to the gradient as in (3.12). We will refer to the resulting algorithm as DP-MASG. The selections in (4.7) for the stage lengths and the stepsizes were designed for constant noise variance per iteration. In the following, we will instead propose a new version that uses a variable noise variance parameter b_t at iteration t, which can improve performance. The main idea is to rely on Proposition 4.2 to optimize over b_t's with the privacy budget constraint.

In order to study how the privacy noise can be optimally distributed to the iterations of DP-MASG, we provide an explicit bound that not only accommodates iteration-dependent noise variance, but also is in the same form as (4.3) so that the noise variances can be optimized to minimize the bound. For MASG, stepsizes change across stages; therefore, the recursion in (4.2) cannot be applied for all iterations. Instead, by Lemma 3.3 of [2], we have a factor of two that appears when the algorithm transitions from one stage to the next. This leads to the following theorem.

THEOREM 4.3. *Let $F \in \mathcal{S}_{\mu,L}(\mathbb{R}^d)$. Consider the DP-MASG algorithm with stage lengths n_k and stepsizes during those stages $\alpha^{(k)}$ given as in* (4.7)*, and with noisy gradients $\nabla f(x_t) + \eta_t$, where $\eta_{t,i} \overset{\text{i.i.d.}}{\sim} \text{Laplace}(b_t)$ for $i = 1, \ldots, d$. Then,*

$$\begin{aligned} E_T = {} & \left[2^{s_T - s_0} \prod_{i=1}^{T} (1 - \sqrt{\mu \alpha^{(s_i)}}) \right] E_0 \\ & + \sum_{t=1}^{T} 2^{s_T - s_t} \left[\prod_{i=t+1}^{T} (1 - \sqrt{\mu \alpha^{(s_i)}}) \right] \alpha^{(s_t)} (1 + \alpha^{(s_t)} L) \left(b_t^2 d + \sigma_s^2(m,n)/2 \right), \end{aligned} \tag{4.8}$$

where s_i is the stage that contains iteration i, provided that $\alpha^{(k)} \leq 1/L$ for all $k \geq 1$.

Observing that the bound in (4.8) is in the same form as in (4.3), b_t can be optimized as in (4.4) but with $a_{T,t}$ indicated by (4.8) as

$$a_{T,t} = 2^{s_T - s_t} \left[\prod_{i=t+1}^{T} (1 - \sqrt{\mu \alpha^{(s_i)}}) \right] \alpha^{(s_t)} (1 + \alpha^{(s_t)} L), \quad t = 1, \ldots, T.$$

Once again, the optimal b_t's when $m = n$ can be written in terms of ϵ, S_1, and $a_{T,t}$'s as in (4.5). To show the effect of algorithm parameters on noise variance, we plot the optimum b_t values in Figure 2 for $\mu = 1$, $L = 20$, $\kappa = 20$, $p = 1$, and $c_1 = 1$, representing the constant factor in front of the stepsize.

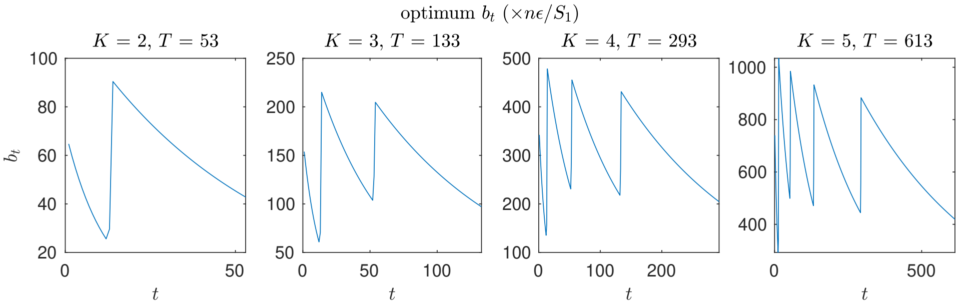

FIG. 2. *Optimal b_t values for the multistage NAG algorithm.*

5. Experimental results. Our experiments concern a regularized logistic regression problem.[2] The model has observations $y_i = (u_i, z_i)$, $i = 1, \ldots, n$, where $u_i \in \mathcal{U} \subseteq \mathbb{R}^d$ is a vector of covariates and $z_i \in \{-1, 1\}$ is a binary response whose conditional probability given u_i depends on a parameter vector $x \in \mathbb{R}^d$ as follows:

$$p(z_i | u_i, x) = \left[1 + e^{-z_i u_i^\top x}\right]^{-1}, \quad i = 1, \ldots, n.$$

Since the probability distribution of u_i's does not depend on x, the (regularized) maximum likelihood problem is defined as determining

$$x^* = \arg\max_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f(x; u_i, z_i), \tag{5.1}$$

where $f(x; u_i, z_i) := \ln p(z_i | u_i, x) + \lambda \|x\|^2$. One can verify that $S_1(x) = 2 \sup_{u \in \mathcal{U}} \|u\|_1$ for all $x \in \mathbb{R}^d$, upon observing that for all $u, u' \in \mathcal{U}$ and $z, z' \in \{0, 1\}^2$, we have

$$\begin{aligned}
\|\nabla f(x; u, z) - \nabla f(x; u', z')\|_1 &= \left\| \frac{zue^{zu^\top x}}{1 + e^{zu^\top x}} - \frac{z'u'e^{zu'^\top x}}{1 + e^{z'u'^\top x}} \right\|_1 \\
&\le \|u\|_1 \left| \frac{ze^{zu^\top x}}{1 + e^{zu^\top x}} \right| + \|u'\|_1 \left| \frac{z'e^{zu'^\top x}}{1 + e^{z'u'^\top x}} \right| \\
&\le \|u\|_1 + \|u'\|_1. \qquad (5.2)
\end{aligned}$$

For the experiments that follow, we use synthetic data with $d = 20$ and $n = 10^5$, and the value of regularization parameter λ is taken as 0.01. The set $\mathcal{U}$ is taken as the set of all $d \times 1$ real-valued vectors with an L_1-norm less than or equal to 20. Hence, Assumption 3.2 holds for this example with $S_1 = 2 \times 20$. We set $\mu = 2 \times \lambda$, and L is estimated as the largest singular value of $\frac{1}{n}(U^\top U) + 2\lambda I_d$, where U is the $n \times d$ matrix with u_t being its column t.

[2] The results are produced with the code at https://github.com/sibirbil/DPAccGradMethods.

In our experiments, we compared six differentially private algorithms. The first four, DP-GD, DP-NAG, DP-MASG, and DP-HB, are the straightforward differentially private versions of GD, NAG, MASG, and HB, respectively. The last two algorithms in the comparison are named DP-NAG-opt and DP-MASG-opt, which stand for the alterations of DP-NAG and DP-MAGS for which the privacy-preserving noise is distributed to the iterations according to Proposition 4.2.

The algorithms are compared across different values of m, T, and c, where m is the subsampling size, T is the number of iterations, and c determines the stepsize as in $\alpha = c/L$. We tried all the combinations (m, T, c) of $m = 10^3, 10^5$, $T = 100, 200, 500, 1000$, and $c = 0.1, 1$. We fixed $\epsilon = 1$ throughout all of the experiments. For DP-MASG and DP-MASG-opt, the general stepsize and stage length formulation in (4.7), presented for the original versions, is preserved; however, the stepsizes are scaled by c. For DP-HB, DP-NAG, and DP-NAG-opt, the momentum parameter is taken as $\beta = (1 - \sqrt{\alpha\mu})/(1 + \sqrt{\alpha\mu})$, while for DP-MASG and DP-MASG-opt the momentum parameter is taken as $\beta^{(k)} = (1 - \sqrt{\alpha^{(k)}\mu})/(1 + \sqrt{\alpha^{(k)}\mu})$ at the kth stage of the algorithm. Finally, the initial point for each algorithm is selected as $x_0 = [10, \dots, 10]^\top$.

For DP-NAG-opt and DP-MASG-opt, we also adjusted the given value of T as follows: With an initial guess of $E_0 = 10$, we computed the bound in (4.6) for each $T' \leq T$, and we decided the number of iterations to be that T' which gives the minimum bound. This procedure was detailed in Remark 4.1.

Figures 3 and 4 show, for $m = 100000$ (no subsampling) and for $m = 1000$, respectively, the performances of the algorithms for the tried values of c and T. Each subfigure shows the log-difference between the objective function evaluated at the current iterate $F(x_t)$ and the objective function evaluated at the optimum solution $F(x^*)$. The optimum solution was found with a nonprivate NAG algorithm that was run for 1000 iterations and without subsampling. The plotted values are the averages from 20 independent runs for each combination of (m, T, c). Trace plots of the iterates for the different values of T are plotted together with different colors. Note that for some cases plots overlap, leaving some colors invisible.

Comparing DP-GD against the accelerated algorithms, we observe that the accelerated algorithms, DP-HB, DP-NAG, and DP-NAG-opt, outperform DP-SGD. Furthermore, among the accelerated algorithms, we have the best results with DP-NAG-opt and DP-MASG-opt. The advantage of accelerating is more striking for $c = 0.1$, representing a too small value for the stepsize. While DP-DG is dramatically slow with a too small stepsize, the accelerated algorithms DP-HB, DP-NAG, and DP-NAG-opt seem to suffer less from that ill choice for the stepsize. When $c = 1$, DP-GD recovers from slow convergence; however, the accelerated algorithms are still able to beat it. Our observations hold both for $m = 100000$ and for $m = 1000$. The multistage algorithm DP-MASG is also prone to a small value for c, but it recovers dramatically when $c = 1$ as recommended in the earlier work [2].

In all instances, we can see the advantage of accelerated algorithms in the speed of convergence. However, when we compare the error levels that the algorithms have reached for the same T, we see that sometimes DP-GD has a better performance than DP-NAG or DP-HB. See, for example, the lower half of Figure 3, at $T = 1000$ (red line): While DP-GD converged more slowly than DP-HB and DP-NAG, it reached a smaller error level. However, if we conduct an *overall* comparison between DP-GD and DP-HB in terms of their best performances among all the choices $T = 100, 200, 500, 1000$, we see that the best of DP-HB (at $T = 100$) outperforms the best of DP-GD (at $T = 1000$). This observation is repeated in our experiments and

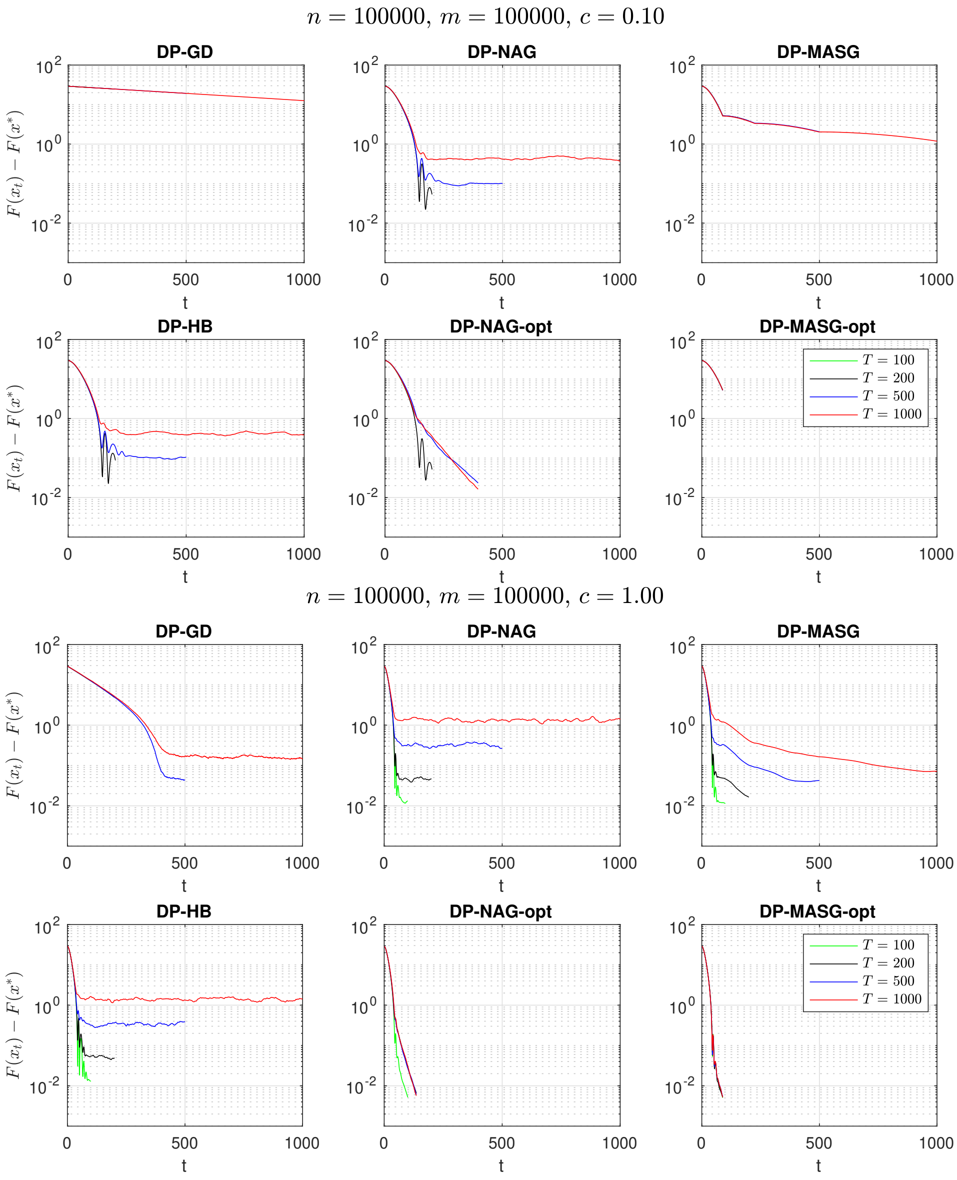

FIG. 3. *Errors with various T and c and without subsampling. Top: $c = 0.1$; bottom: $c = 1$. Note that the plots consist of all the algorithms performed with all the given iteration numbers although some lines are not clearly visible because of overlapping. (Color available online.)*

is suggestive of a general recommendation: The accelerated algorithms can promise faster convergence when used with a small number of iterations.

This general recommendation about the selection of T is further supported by the traces belonging to DP-NAG-opt and DP-MASG-opt (when $c = 1$), where T is readjusted according to (4.6). We can see from the subplots belonging to DP-NAG-opt and DP-MASG-opt that the readjustment prefers small T, and this selection indeed improves the performance. This further justifies the use of the optimized algorithms DP-NAG-opt and DP-MASG-opt, where the distribution of privacy-preserving vari-

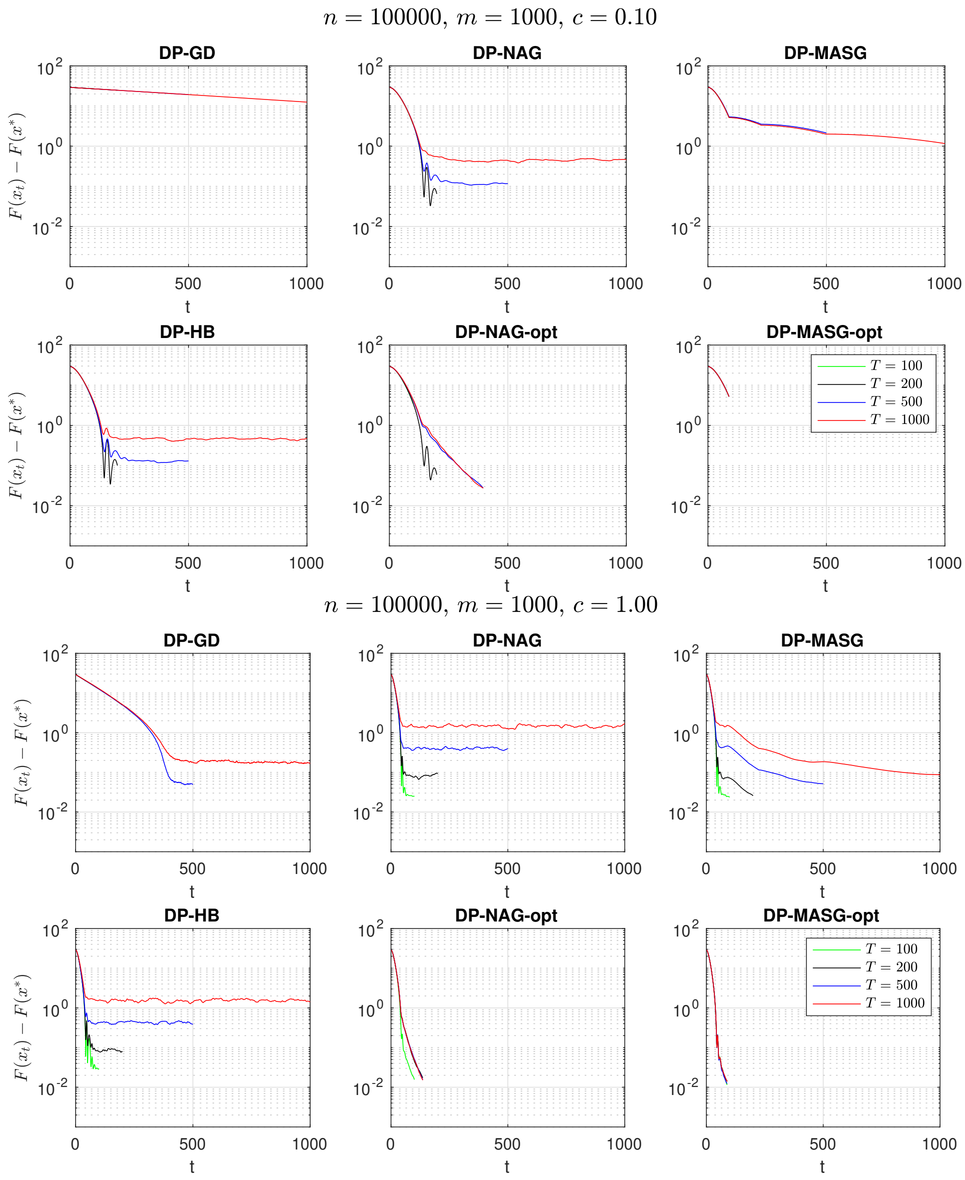

FIG. 4. *Errors with various T and c and with $m = 10^3$. Top: $c = 0.1$; bottom: $c = 1$. Note that the plots consist of all the algorithms performed with all the given iteration numbers although some lines are not clearly visible because of overlapping. (Color available online.)*

ance as well as the number of iterations are chosen automatically.

We also compare between the NAG-based schemes and their multistage versions. When the stepsize is chosen properly (cf. $c = 1$), both DP-NAG-opt and DP-MASG-opt perform very similarly and outperform the others. However, DP-NAG-opt seems more robust to a poor selection of the stepsize (as shown for $c = 0.1$).

Finally, we compare the selections $m = 1000$ and $m = 100000$, where the first one corresponds to subsampling (with a rate of 1%), and the other corresponds to no subsampling. First, we can see that, even when we subsample, optimizing b_t's and T according to Proposition 4.2 does improve the performance of DP-NAG and DP-MASG significantly. (Recall that Proposition 4.2 holds under no subsampling, yet its use under subsampling was discussed in Remark 4.2.) Second, a comparison of Figures 3 and 4 on the whole shows that using full data improves the performance of the accelerated algorithms, especially for $c = 1$ (compare the lower halves of the figures). However, the difference does not seem to be by an order of magnitude. Since the additional randomness introduced by subsampling helps to decrease the required noise level for DP and using a sample instead of full data at each iteration is faster (in terms of per iteration running time), many DP methods in the literature consider stochastic algorithms in the DP context, where stochastic algorithms can improve the running time compared to deterministic algorithms. However, if the running time is not of concern for reaching a given privacy level, our experiments show that using full data results in a smaller bound on the objective error.

6. Conclusions. In this paper, we presented two classes of differentially private optimization algorithms based on the heavy ball method and Nesterov's accelerated gradient method. We provided performance bounds for our algorithms for a given iteration budget while preserving a desired privacy level depending on the choice of the parameters (stepsize and the momentum). We showed that, for NAG, homogeneous distribution of the privacy budget over all iterations, as typically done in the literature so far, is not the best way, and we propose a method to improve it. Numerical experiments showed that the presented algorithms have advantages over their well-known straightforward versions.

Our analysis and methodology can be adapted to other forms of privacy to a certain extent. For this, existence of a tractable formula for the noise parameter to satisfy a certain level of privacy is the key requirement. For example, a weaker form of (ϵ, δ)-differential privacy can be satisfied if the normal distribution is used for the privacy-preserving noise and the required noise variance is well known [13]. Furthermore, provided there is no subsampling, privacy loss can be optimally distributed to the iterations of DP-NAG using a closed-form formula, as in Proposition 4.2, by exploiting the relation between zero-concentrated differential privacy of [7] and (ϵ, δ)-differential privacy.

Our theoretical work formally investigates, for DP-HB and DP-NAG, as well as the multistage version of DP-NAG, the effect of the algorithm parameters on the error bound. For DP-NAG and its multistage version, we also provide explicit formulas about how the variance of the gradient noise should be tuned at each stage to preserve a certain given level of privacy requirement, given the choice for the total number of iterations. However, in our setup, tuning of these parameters requires knowledge about the constants μ and L. The Lipschitz constant L can often be estimated from data using line search techniques (see, e.g., [37, Algorithm 2] or [6]). The strong convexity constant μ may also be known in some cases; for instance, if a regularization term $\lambda\frac{\|x\|^2}{2}$ with $\lambda > 0$ is added to a convex empirical risk minimization problem of the form (2.2), the strong convexity constant μ can be taken as λ. However, in general, μ may not be known, and it may need to be estimated from data. As part of future work, it would be interesting to investigate whether restarting techniques developed for accelerated deterministic algorithms such as [17] which do not require knowledge of the strong convexity constant a priori can be adapted to the privacy setting.

Appendix A. Definitions and known results.

DEFINITION A.1 (strongly convex and smooth functions). *A continuously differ-*

entiable function $F : \mathbb{R}^d \mapsto \mathbb{R}$ is called strongly convex with modulus $\mu > 0$ and L-smooth with a Lipschitz constant $L > 0$ if it satisfies

$$\frac{\mu}{2}\|x-y\|^2 \le F(x) - F(y) - \nabla F(y)^\top (x-y) \le \frac{L}{2}\|x-y\|^2 \quad \forall x, y \in \mathbb{R}^d.$$

The inequalities on the left- and right-hand sides define strong convexity and L-smoothness, respectively. Moreover, $\mathcal{S}_{\mu,L}(\mathbb{R}^d)$ denotes the set of continuously differentiable functions that are strongly convex with modulus μ and L-smooth with L.

Definition A.2 (ϵ-differential privacy [12, Definition 1]). *A randomized algorithm $\mathcal{A}$ with set of input datasets $\mathcal{Y}$ and range for its output $\mathcal{X}$ is ϵ-differentially private if for all datasets $Y, Y' \in \mathcal{Y}$ differing by at most one element, i.e., $h(Y, Y') \le 1$, and all measurable $O \subseteq \mathcal{X}$, it holds that $\mathbb{P}[A_Y \in O] \le e^\epsilon \mathbb{P}[A_{Y'} \in O]$.*

Definition A.3 (sensitivity [13, Definition 3.1]). *For a function on datasets $\varphi : \mathcal{Y} \mapsto \mathbb{R}^k$, $k \ge 1$, the L_1-sensitivity of φ is defined as*

$$S_1^\varphi = \max_{Y,Y' \in \mathcal{Y} : h(Y,Y')=1} ||\varphi(Y) - \varphi(Y')||_1. \tag{A.1}$$

Theorem A.1 (Laplace mechanism [12, Theorem 1]). *Given function $\varphi : \mathcal{Y} \mapsto \mathbb{R}^k$, the mechanism $\mathcal{A}_\varphi$, which adds independently generated noise with Laplace distribution* $\mathrm{Lap}(S_1^\varphi/\epsilon)$ *to each of the k output terms, is ϵ-differentially private.*

The methods that we shall present in the subsequent sections use the Laplace mechanism at every iteration. Thus, we further need to quantify the privacy loss due to using a randomized algorithm repeatedly.

Theorem A.2 (composition [13, Corollary 3.15]). *Let each algorithm $\mathcal{A}_i$ be ϵ_i-differentially private. Then $(\mathcal{A}_1, \dots, \mathcal{A}_T)$, whose output is the concatenation of the outputs of the individual algorithms, is $\sum_{i=1}^T \epsilon_i$-differentially private.*

Theorem A.3 ([4, Theorem 9]). *Let $\mathcal{M} : \bigcup_{i=1}^\infty \mathcal{Y}^n \to \mathcal{X}$ be an ϵ-differentially private algorithm, and let the elements of $\mathcal{Y}^n$ be in the form of $y_{1:n}$. Then, an algorithm $\mathcal{M}_{m,n} : \mathcal{Y}^n \to \mathcal{X}$ that first selects a random subsample of m items from its input data $y_{1:n} \in \mathcal{Y}^n$, by sampling without replacement, and then runs $\mathcal{M}$ on the subsample is ϵ'-differentially private, where $\epsilon' = \ln\left(1 + \frac{m}{n}(e^\epsilon - 1)\right)$.*

Appendix B. Omitted proofs. We reserve this section for the proofs of several results in the main text.

B.1. Proof of Proposition 3.3.

Proof. Fix $x_t = x \in \mathbb{R}^d$ for the rest of the proof. Since $\mathrm{Cov}(\eta_t|x) = 2b_t(x)^2 I_d$, where $b_t(x) = S_1(x)/m\epsilon_0$, Assumption 3.2 implies that

$$\|\mathrm{Cov}(\eta_t|x)\| = \frac{S_1(x)^2 d}{m^2\epsilon_0^2} \le \frac{S_1^2 d}{m^2 \epsilon_0^2}. \tag{B.1}$$

Next, let $R = \mathrm{Cov}\left(\nabla F_B(x)|x\right)$, where $\nabla F_B(x)$ is the subsampling-based estimator of $\nabla F(x)$ when the indices in the subsample B are sampled without replacement. From the unbiasedness property of $\nabla F_B(x)$, we have $R = \mathrm{Cov}(v_t|x)$. The diagonal terms in $R = [r_{i,j}]$ can be written as

$$r_{k,k} = \sigma_{y,k}^2 \frac{1}{m}\frac{n-m}{n-1}, \quad k = 1, \dots, d, \tag{B.2}$$

where $\sigma^2_{y,k}$ is the population variance given by

$$\sigma^2_{y,k} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\partial f(x;y_i)}{\partial x_k} - \frac{1}{n}\sum_{j=1}^{n}\frac{\partial f(x;y_j)}{\partial x_k}\right)^2. \tag{B.3}$$

Let

$$S_{1,k}(x) = \sup_{y,y'\in\mathcal{Y}}\left|\frac{\partial f(x;y)}{\partial x_k} - \frac{\partial f(x;y')}{\partial x_k}\right|, \quad k = 1,\dots,d.$$

The population variance in (B.3) can then be bounded as $\sigma^2_{y,k} \le S_{1,k}(x)^2/4$. Therefore, we bound $\|R\|$ by its trace as

$$\begin{aligned}\|R\| &\le \frac{1}{4}\left[\sum_{k=1}^{d} S_{1,k}(x)^2\right]\frac{1}{m}\frac{n-m}{n-1}\\ &\le \frac{1}{4}\left[\sum_{k=1}^{d} S_{1,k}(x)\right]^2\frac{1}{m}\frac{n-m}{n-1} \le \frac{1}{4}S_1^2\frac{1}{m}\frac{n-m}{n-1},\end{aligned} \tag{B.4}$$

where the last line follows from Assumption 3.2. Combining (B.1) and (B.4) and using the triangle inequality for the matrix norm, we have the claimed bound on $||\mathrm{Cov}(w_t|x)||$. □

B.2. Proof of Proposition 3.4. Recall, from section 3.1, the dynamic system representation in (3.9) and (3.10), and define $\bar{F} : \mathbb{R}^{2d} \mapsto \mathbb{R}$ such that for $\xi_t = \begin{bmatrix} x_t^\top & x_{t-1}^\top\end{bmatrix}^\top$ we have $\bar{F}(\xi_t) = F(x_t)$. Also, with X_1, X_2 defined in Proposition 3.4, we define $\tilde{X}_1 = X_1 \otimes I_d$ and $\tilde{X}_2 = X_2 \otimes I_d$.

The following lemma is central to the proof of Proposition 3.4.

LEMMA B.1. *Let $F \in \mathcal{S}_{\mu,L}(\mathbb{R}^d)$, and consider the DP-SHB algorithm. Let $w_t = \eta_t + v_t$, the overall noise added to $\nabla F(x_t)$ due to the Laplace mechanism and subsampling. Then for any $\rho \in (0, 1)$, we have*

$$\mathbb{E}\left[\begin{bmatrix}\xi_t - \xi^*\\ \nabla F(z_t)\end{bmatrix}^\top (\tilde{X}_1 + (1-\rho^2)\tilde{X}_2)\begin{bmatrix}\xi_t - \xi^*\\ \nabla F(z_t)\end{bmatrix}\right] \le \rho^2\mathbb{E}[\bar{F}(\xi_t) - F^*] - \mathbb{E}[\bar{F}(\xi_{t+1}) - F^*] + \frac{L\alpha^2}{2}\mathbb{E}[\|w_t\|^2].$$

Proof. One update rule of DP-SBH can be rewritten as

$$x_{t+1} = (1+\beta)x_t - \beta x_{t-1} - \alpha(\nabla F(x_t) + w_t). \tag{B.5}$$

Using (B.5), we have

$$\begin{aligned} x_t - x_{t+1} &= x_t - (1+\beta)x_t + \beta x_{t-1} + \alpha(\nabla F(x_t) + w_t)\\ &= \beta(-x_t + x_{t-1}) + \alpha(\nabla F(x_t) + w_t).\end{aligned} \tag{B.6}$$

Since $F \in \mathcal{S}_{\mu,L}(\mathbb{R}^d)$, from Definition A.1, using the inequality on the L-smoothness of F, we can write

$$F(x_t) - F(x_{t+1}) \ge \nabla F(x_t)^\top(x_t - x_{t+1}) - \frac{L}{2}\|x_{t+1} - x_t\|^2. \tag{B.7}$$

Combining (B.7) with (B.6), we obtain

$$\begin{aligned}F(x_t)-F(x_{t+1}) \geq{}& \nabla F(x_t)^\top(-\beta(x_t-x_{t-1})+\alpha(\nabla F(x_t)+w_t))\\&-\frac{L}{2}\|\beta(x_t-x_{t-1})-\alpha(\nabla F(x_t)+w_t)\|^2\\={}&-\beta(x_t-x_{t-1})^\top\nabla F(x_t)+\alpha\|\nabla F(x_t)\|^2+\alpha\nabla F(x_t)^\top w_t\\&-\frac{L\beta^2}{2}\|x_t-x_{t-1}\|^2+L\alpha\beta(x_t-x_{t-1})^\top(\nabla F(x_t)+w_t)\\&-\frac{\alpha^2L}{2}\|\nabla F(x_t)+w_t\|^2\\={}&\frac{1}{2}\begin{bmatrix}x_t-x_{t-1}\\\nabla F(x_t)\end{bmatrix}^\top\tilde{D}\begin{bmatrix}x_t-x_{t-1}\\\nabla F(x_t)\end{bmatrix}-\frac{L\alpha^2}{2}\|w_t\|^2+L\alpha[\beta(x_t-x_{t-1})\\&-\alpha\nabla F(x_t)]^\top w_t+\alpha\nabla F(x_t)^\top w_t,\end{aligned}$$

where $\tilde{D}=D\otimes I_d$ is a $2d\times 2d$ matrix defined through

$$D=\begin{bmatrix}-L\beta^2 & L\alpha\beta-\beta\\L\alpha\beta-\beta & -\alpha^2L+2\alpha\end{bmatrix}.$$

Next, note that

$$\begin{bmatrix}x_t-x_{t-1}\\\nabla F(h_t)\end{bmatrix}=\begin{bmatrix}I_d & -I_d & 0_d\\0_d & 0_d & I_d\end{bmatrix}\begin{bmatrix}x_t-x^*\\x_{t-1}-x^*\\\nabla F(x_t)\end{bmatrix}$$

and

$$\frac{1}{2}\begin{bmatrix}I_d & -I_d & 0_d\\0_d & 0_d & I_d\end{bmatrix}^\top\tilde{D}\begin{bmatrix}I_d & -I_d & 0_d\\0_d & 0_d & I_d\end{bmatrix}=\tilde{X}_1.$$

Thus,

$$\begin{aligned}F(x_t)-F(x_{t+1})\geq{}&\begin{bmatrix}x_t-x^*\\x_{t-1}-x^*\\\nabla F(x_t)\end{bmatrix}^\top\tilde{X}_1\begin{bmatrix}x_t-x^*\\x_{t-1}-x^*\\\nabla F(x_t)\end{bmatrix}\\&-\frac{L\alpha^2}{2}\|w_t\|^2+L\alpha\left[\beta(x_t-x_{t-1})-\alpha\nabla F(x_t)\right]^\top w_t-\alpha\nabla F(x_t)^\top w_t.\end{aligned}\tag{B.8}$$

Similarly, by the inequality that gives strong convexity in Definition A.1, we have

$$\begin{aligned}F(x^*)-F(x_t)\geq{}&\nabla F(x_t)^\top(x^*-x_t)+\frac{\mu}{2}\|x^*-x_t\|^2\\={}&\frac{1}{2}\begin{bmatrix}x_t-x^*\\x_{t-1}-x^*\\\nabla F(x_t)\end{bmatrix}^\top\begin{bmatrix}\mu I_d & 0_d & -I_d\\0_d & 0_d & 0_d\\-I_d & 0_d & 0_d\end{bmatrix}\begin{bmatrix}x_t-x^*\\x_{t-1}-x^*\\\nabla F(x_t)\end{bmatrix}.\end{aligned}$$

The matrix in the middle is equal to $\tilde{X}_2$, so we can write

$$F(x^*)-F(x_t)\geq\frac{1}{2}\begin{bmatrix}x_t-x^*\\x_{t-1}-x^*\\\nabla F(x_t)\end{bmatrix}^\top\tilde{X}_2\begin{bmatrix}x_t-x^*\\x_{t-1}-x^*\\\nabla F(x_t)\end{bmatrix}.\tag{B.9}$$

Multiplying (B.9) by $(1-\rho^2)$ and adding to (B.8), we obtain

$$\begin{aligned}\begin{bmatrix}\xi_t-\xi^*\\ \nabla F(x_t)\end{bmatrix}^\top [\tilde{X}_1+(1-\rho^2)\tilde{X}_2]\begin{bmatrix}\xi_t-\xi^*\\ \nabla F(x_t)\end{bmatrix} &\le \rho^2[F(x_t)-F^*]-(F(x_{t+1})-F^*)\\ &+\frac{L\alpha^2}{2}\|w_t\|^2-L\alpha\left[\beta(x_t-x_{t-1})-\alpha\nabla F(x_t)\right]w_t-\alpha\nabla F(x_t)w_t.\end{aligned}$$

Taking the expectation and applying $\mathbb{E}(w_t)=0$, we have the desired result. ∎

Proof of Proposition 3.4. Lemma B.1 is the counterpart of Lemma 4.5 of [3], which is given for NAG. Hence, Lemma B.1 allows us to extend the NAG results in [3, Proposition 4.6 and Corollary 4.7] for DP-SHB. Finally, under Assumption 3.2, we get the desired bound in our proposition. ∎

B.3. Proof of Theorem 3.5.

Proof. Following the proof technique of [8, Theorem 12], we can write

$$\text{(B.10)}\qquad \begin{bmatrix}x_t-x^*\\ x_{t-1}-x^*\end{bmatrix}=M(\alpha,\beta)\begin{bmatrix}x_{t-1}-x^*\\ x_{t-2}-x^*\end{bmatrix}+\begin{bmatrix}-\alpha w_t\\ 0_d\end{bmatrix},$$

where we have

$$M(\alpha,\beta)=\begin{bmatrix}(1+\beta)I_d-\alpha Q & -\beta I_d\\ I_d & 0_d\end{bmatrix}.$$

There also exists a permutation matrix P such that

$$PM(\alpha,\beta)P^\top=\bar{T}:=\begin{bmatrix}T_1 & \cdots & 0 & 0\\ 0 & T_2 & \cdots & 0\\ \vdots & \cdots & \ddots & \vdots\\ 0 & 0 & \cdots & T_d\end{bmatrix},$$

where

$$T_i=\begin{bmatrix}1+\beta-\alpha\lambda_i & -\beta\\ 1 & 0\end{bmatrix},\quad 1\le i\le d,$$

are 2×2 matrices with eigenvalues

$$a_{\lambda_i,\pm}=\frac{1+\beta-\alpha\lambda_i\pm\sqrt{(1+\beta-\alpha\lambda_i)^2-4\beta}}{2}.$$

Therefore, for $t\ge 1$ we obtain

$$\left\|M(\alpha,\beta)^t\right\|=\left\|P^\top\bar{T}^tP\right\|\le\|P^\top\|\|P\|\max_{1\le i\le d}\left\|T_i^t\right\|=\max_{1\le i\le d}\left\|T_i^t\right\|,$$

where we used the fact that $\|P\|=1$ for a permutation matrix P. T_i^t is a 2×2 matrix; it has either semisimple eigenvalues or a defective eigenvalue with a multiplicity two. In either case, it is well known that we can write $\|T_i^t\|\le C_i^t\rho_{\lambda_i}^t$, where $\rho_{\lambda_i}=\max\{|a_{\lambda_i,+}|,\|a_{\lambda_i,-}\|\}$ is the spectral radius of T_i^t and $C_i^t=\mathcal{O}(t)$. Then it follows that $\|M(\alpha,\beta)^t\|\le C_t\rho^t$, where we take $C_t=\max_i\{C_i^t\}$ and $\rho=\max_i\{\rho_{\lambda_i}\}$. After a straightforward computation, we observe that ρ_λ is a quasi-convex function of λ; therefore the function ρ_λ attains its maximum as a function of λ on the interval $[\mu,L]$ for either $\lambda=\mu$ or $\lambda=L$. Thus, ρ can also be written as

$$\rho=\max\{\rho_{\lambda_\mu},\rho_{\lambda_L}\}=\max\{|a_{\mu,+}|,|a_{\mu,-}|,|a_{L,+}|,|a_{L,-}|\}.$$

Let $\hat{E}_t = \mathbb{E}\left[(\xi_t - \xi_*)(\xi_t - \xi_*)^\top | \xi_{t-1}\right]$. From (B.10), we obtain the recursion

$$\begin{aligned}\hat{E}_{t+1} &= M(\alpha,\beta)\left[(\xi_t - \xi_*)(\xi_t - \xi_*)^\top\right] M^\top(\alpha,\beta) + \begin{pmatrix} \alpha^2 \mathbb{E}(w_t w_t^\top | x_t) & 0_d \\ 0_d & 0_d \end{pmatrix} \\ &\preceq M(\alpha,\beta)\left[(\xi_t - \xi_*)(\xi_t - \xi_*)^\top\right] M^\top(\alpha,\beta) + \begin{pmatrix} \alpha^2 \Sigma & 0_d \\ 0_d & 0_d \end{pmatrix}.\end{aligned}$$

Taking expectations with respect to ξ_t, we find

$$\bar{E}_{t+1} \preceq M(\alpha,\beta)\bar{E}_t M^\top(\alpha,\beta) + \begin{pmatrix} \alpha^2 \sigma_T^2 I & 0_d \\ 0_d & 0_d \end{pmatrix},$$

where we let $\bar{E}_t := \mathbb{E}\left[(\xi_t - \xi_*)(\xi_t - \xi_*)^\top\right]$. We can also write

$$\begin{aligned}\operatorname{Tr}\left(\bar{E}_t\right) &\leq m(\alpha,\beta) + (M(\alpha,\beta))^t \bar{E}_0 \left(M(\alpha,\beta)^\top\right)^t \\ &\quad - \sum_{j=t}^{\infty} M(\alpha,\beta)^j \begin{pmatrix} \alpha^2 c_W I & 0_d \\ 0_d & 0_d \end{pmatrix} \left(M(\alpha,\beta)^\top\right)^j \\ &\leq m(\alpha,\beta) + \left\| M(\alpha,\beta)^t \right\|^2 \bar{E}_0 + \sum_{j=t}^{\infty} \left\| M(\alpha,\beta)^j \right\|^2 \alpha^2 \|\Sigma\| \\ &\leq m(\alpha,\beta) + C_t^2 \rho^{2t} \bar{E}_0 + \alpha^2 \sigma_T^2 C_t^2 \frac{\rho^{2t}}{1-\rho^2},\end{aligned}$$

where we used the estimate $\|M(\alpha,\beta)^t\| \leq C_t \rho^t$. This completes the proof. □

B.4. Proof of Proposition 4.2.

Proof. Observe from (3.3) that, for $m = n$, we have $\varepsilon(S_1, b_t, n, n) = S_1/(b_t n)$. Hence, the optimization problem in (4.4) reduces to minimizing $\sum_{t=1}^T a_{T,t} b_t^2$ over $b_1, \ldots, b_T$ subject to $\sum_{t=1}^T \frac{S_1}{n b_t} = \epsilon$. The above optimization problem can be solved by equating the gradient of the corresponding Lagrangian function $\sum_{t=1}^T a_{T,t} b_t^2 + \lambda\big(\sum_{t=1}^T S_1/(n b_t) - \epsilon\big)$ with respect to $(b_1, \ldots, b_T, \lambda)$ to 0, which yields the system of $T+1$ equations $2 a_{T,t} b_t = \frac{\lambda S_1}{n b_t^2}$ for $t = 1, \ldots, T$ and $\sum_{t=1}^T \frac{S_1}{n b_t} = \epsilon$, which has the solution

$$b_t = \left(\frac{\lambda S_1/n}{2 a_{T,t}}\right)^{1/3}, \quad \text{with} \quad \lambda = \frac{(S_1/n)^2 \left[\sum_{t=1}^T (2 a_{T,t})^{1/3}\right]^3}{\epsilon^3}.$$

Substituting λ into b_t yields the claimed solution. Finally, the bordered Hessian at the solution is a diagonal matrix, with T negative values and a single 0 on its diagonal. □

REFERENCES

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, *Deep learning with differential privacy*, in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2016, pp. 308–318.

[2] N. S. Aybat, A. Fallah, M. Gürbüzbalaban, and A. Ozdaglar, *A universally optimal multistage accelerated stochastic gradient method*, in Advances in Neural Information Processing Systems, 2019, pp. 8525–8536.

[3] N. S. Aybat, A. Fallah, M. Gürbüzbalaban, and A. Ozdaglar, *Robust accelerated gradient methods for smooth strongly convex functions*, SIAM J. Optim., 30 (2020), pp. 717–751, https://doi.org/10.1137/19M1244925.

[4] B. Balle, G. Barthe, and M. Gaboardi, *Privacy amplification by subsampling: Tight analyses via couplings and divergences*, in Advances in Neural Information Processing Systems, 2018, pp. 6277–6287.

[5] R. Bassily, A. Smith, and A. Thakurta, *Private empirical risk minimization: Efficient algorithms and tight error bounds*, in 2014 IEEE 55th Annual Symposium on Foundations of Computer Science, IEEE, 2014, pp. 464–473.

[6] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202, https://doi.org/10.1137/080716542.

[7] M. Bun and T. Steinke, *Concentrated differential privacy: Simplifications, extensions, and lower bounds*, in Proceedings, Part I, of the 14th International Conference on Theory of Cryptography - Volume 9985, Springer-Verlag, New York, 2016, pp. 635–658.

[8] B. Can, M. Gürbüzbalaban, and L. Zhu, *Accelerated linear convergence of stochastic momentum methods in Wasserstein distances*, in Proceedings of the 36th International Conference on Machine Learning, 2019.

[9] K. Chaudhuri and C. Monteleoni, *Privacy-preserving logistic regression*, in Advances in Neural Information Processing Systems, 2009, pp. 289–296.

[10] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, *Differentially private empirical risk minimization*, J. Mach. Learn. Res., 12 (2011), pp. 1069–1109.

[11] C. Dwork, *Differential privacy*, in 33rd International Colloquium on Automata, Languages and Programming (ICALP 2006), Part II, 2006, Springer-Verlag.

[12] C. Dwork, *Differential privacy: A survey of results*, in International Conference on Theory and Applications of Models of Computation, Springer, 2008, pp. 1–19.

[13] C. Dwork and A. Roth, *The algorithmic foundations of differential privacy*, Found. Trends Theor. Comput. Sci., 9 (2014), pp. 211–407.

[14] C. Dwork and G. N. Rothblum, *Concentrated Differential Privacy*, preprint, https://arxiv.org/abs/1603.01887, 2016.

[15] C. Dwork, G. N. Rothblum, and S. Vadhan, *Boosting and differential privacy*, in 2010 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE, 2010, pp. 51–60.

[16] M. Fazlyab, A. Ribeiro, M. Morari, and V. M. Preciado, *Analysis of optimization algorithms via integral quadratic constraints: Nonstrongly convex problems*, SIAM J. Optim., 28 (2018), pp. 2654–2689, https://doi.org/10.1137/17M1136845.

[17] O. Fercoq and Z. Qu, *Adaptive restart of accelerated gradient methods under local quadratic growth condition*, IMA J. Numer. Anal., 39 (2019), pp. 2069–2095.

[18] N. Flammarion and F. Bach, *From averaging to acceleration, there is only a step-size*, in Proceedings of the 28th Conference on Learning Theory, PMLR 40, 2015, pp. 658–695.

[19] J. Foulds, J. Geumlek, M. Welling, and K. Chaudhuri, *On the theory and practice of privacy-preserving Bayesian data analysis*, in Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, UAI'16, Arlington, VA, 2016, AUAI Press, pp. 192–201.

[20] S. Gadat, F. Panloup, and S. Saadane, *Stochastic heavy ball*, Electron. J. Stat., 12 (2018), pp. 461–529.

[21] B. Hu and L. Lessard, *Dissipativity theory for Nesterov's accelerated method*, in Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR.org, 2017, pp. 1549–1557.

[22] S. L. Hyland and S. Tople, *On the Intrinsic Privacy of Stochastic Gradient Descent*, preprint, https://arxiv.org/abs/1912.02919, 2019.

[23] B. Jayaraman, L. Wang, D. Evans, and Q. Gu, *Distributed learning without distress: Privacy-preserving empirical risk minimization*, in Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, Curran Associates Inc., Red Hook, NY, 2018, pp. 6346–6357.

[24] D. Kifer, A. Smith, and A. Thakurta, *Private convex empirical risk minimization and high-dimensional regression*, in Proceedings of the 25th Conference on Learning Theory, PMLR 23, 2012, pp. 25.1–25.40.

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

[26] L. Lessard, B. Recht, and A. Packard, *Analysis and design of optimization algorithms via integral quadratic constraints*, SIAM J. Optim., 26 (2016), pp. 57–95, https://doi.org/10.1137/15M1009597.

[27] N. Loizou and P. Richtárik, *Momentum and stochastic momentum for stochastic gradient, Newton, proximal point and subspace descent methods*, Comput. Optim. Appl., 77 (2020), pp. 653–710.

[28] I. Mironov, *Rényi differential privacy*, in 2017 IEEE 30th Computer Security Foundations Symposium (CSF), 2017, pp. 263–275.

[29] H. Mohammadi, M. Razaviyayn, and M. R. Jovanović, *Robustness of accelerated first-order algorithms for strongly convex optimization problems*, IEEE Trans. Automat. Control, 66 (2021), pp. 2480–2495.

[30] Y. Nesterov, *A method of solving a convex programming problem with convergence rate* $O(1/k^2)$, Soviet Math. Dokl., 27 (1983), pp. 372–376.

[31] M. Park, J. Foulds, K. Chaudhuri, and M. Welling, *Variational Bayes in private settings (VIPS)*, J. Artificial Intelligence Res., 68 (2020), pp. 109–157.

[32] V. Pichapati, A. T. Suresh, F. X. Yu, S. J. Reddi, and S. Kumar, *AdaCliP: Adaptive Clipping for Private SGD*, preprint, https://arxiv.org/abs/1908.07643, 2019.

[33] B. T. Polyak, *Some methods of speeding up the convergence of iteration methods*, U.S.S.R. Comput. Math. and Math. Phys., 4 (1964), pp. 1–17.

[34] B. T. Polyak, *Introduction to Optimization*, Vol. 1, Optimization Software, Publications Division, New York, 1987.

[35] A. Ramezani-Kebrya, A. Khisti, and B. Liang, *On the Stability and Convergence of Stochastic Gradient Descent with Momentum*, preprint, https://arxiv.org/abs/1809.04564, 2018.

[36] B. I. P. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft, *Learning in a Large Function Space: Privacy-Preserving Mechanisms for SVM Learning*, preprint, https://arxiv.org/abs/0911.5708, 2009.

[37] M. Schmidt, R. Babanezhad, M. Ahmed, A. Defazio, A. Clifton, and A. Sarkar, *Non-uniform stochastic average gradient method for training conditional random fields*, in Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, PMLR 38, 2015, pp. 819–828.

[38] R. Shokri and V. Shmatikov, *Privacy-preserving deep learning*, in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, ACM, 2015, pp. 1310–1321.

[39] S. Song, O. Thakkar, and A. Thakurta, *Characterizing Private Clipped Gradient Descent on Convex Generalized Linear Problems*, preprint, https://arxiv.org/abs/2006.06783, 2020.

[40] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Science & Business Media, 2013.

[41] D. Wang, M. Ye, and J. Xu, *Differentially private empirical risk minimization revisited: Faster and more general*, in Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Red Hook, NY, Curran Associates Inc., 2017, pp. 2719–2728.

[42] L. Wang and Q. Gu, *Differentially private iterative gradient hard thresholding for sparse learning*, in Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI'19, AAAI Press, 2019, pp. 3740–3747.

[43] Y. Yan, T. Yang, Z. Li, Q. Lin, and Y. Yang, *A Unified Analysis of Stochastic Momentum Methods for Deep Learning*, preprint, https://arxiv.org/abs/1808.10396, 2018.

[44] T. Yang, Q. Lin, and Z. Li, *Unified Convergence Analysis of Stochastic Momentum Methods for Convex and Non-convex Optimization*, preprint, https://arxiv.org/abs/1604.03257, 2016.

[45] L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex, *Differentially private model publishing for deep learning*, in 2019 IEEE Symposium on Security and Privacy (SP), IEEE, 2019, pp. 332–349.

[46] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, *Functional mechanism: Regression analysis under differential privacy*, Proc. VLDB Endow., 5 (2012), pp. 1364–1375.

[47] J. Zhang, K. Zheng, W. Mou, and L. Wang, *Efficient Private ERM for Smooth Objectives*, preprint, https://arxiv.org/abs/1703.09947, 2017.