# A Hybrid Stochastic-Deterministic Minibatch Proximal Gradient Method for Efficient Optimization and Generalization

Pan Zhou, Xiao-Tong Yuan, *Member, IEEE,* Zhouchen Lin, *Fellow, IEEE,* Steven C.H. Hoi, *Fellow, IEEE*

**Abstract**—Despite the success of stochastic variance-reduced gradient (SVRG) algorithms in solving large-scale problems, their stochastic gradient complexity often scales linearly with data size and is expensive for huge data. Accordingly, we propose a hybrid stochastic-deterministic minibatch proximal gradient (HSDMPG) algorithm for strongly convex problems with linear prediction structure, e.g. least squares and logistic/softmax regression. HSDMPG enjoys improved computational complexity that is data-size-independent for large-scale problems. It iteratively samples an evolving minibatch of individual losses to estimate the original problem, and can efficiently minimize the sampled subproblems. For strongly convex loss of $n$ components, HSDMPG attains an $\epsilon$-optimization-error within $\mathcal{O}\left(\kappa \log^{\zeta+1}\left(\frac{1}{\epsilon}\right)\frac{1}{\epsilon}\bigwedge n\log^{\zeta}\left(\frac{1}{\epsilon}\right)\right)$ stochastic gradient evaluations, where $\kappa$ is condition number, $\zeta=1$ for quadratic loss and $\zeta=2$ for generic loss. For large-scale problems, our complexity outperforms those of SVRG-type algorithms with/without dependence on data size. Particularly, when $\epsilon=\mathcal{O}(1/\sqrt{n})$ which matches the intrinsic excess error of a learning model and is sufficient for generalization, our complexity for quadratic and generic losses is respectively $\mathcal{O}(n^{0.5}\log^{2}(n))$ and $\mathcal{O}(n^{0.5}\log^{3}(n))$, which for the first time achieves optimal generalization in less than a single pass over data. Besides, we extend HSDMPG to online strongly convex problems and prove its higher efficiency over the prior algorithms. Numerical results demonstrate the computational advantages of HSDMPG.

**Index Terms**—Convex Optimization, Precondition, Online Convex Optimization, Stochastic Variance-Reduced Algorithm

## 1 Introduction

Convex optimization problem has received broad interests, because it has applications in a wide range of disciplines, such as computer vision [1]–[5], signal processing [6]–[8], statistical learning [9]–[11], finance [12], [13]. In this work, we are particularly interested in the following *finite-sum* or *online* strongly convex optimization problems with linear prediction structure:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} F(\boldsymbol{\theta}) =: \begin{cases} \frac{1}{n}\sum_{i=1}^{n}\ell(\boldsymbol{\theta}^\top\boldsymbol{x}_i,\boldsymbol{y}_i)+\frac{\tau\mu}{2}\|\boldsymbol{\theta}\|_2^2 & \text{(finite-sum)}\\ \mathbb{E}[\ell(\boldsymbol{\theta}^\top\boldsymbol{x},\boldsymbol{y};\pi)]+\frac{\tau\mu}{2}\|\boldsymbol{\theta}\|_2^2 & \text{(online)}\end{cases}, \tag{1}$$

where the convex loss function $\ell(\boldsymbol{\theta}^\top\boldsymbol{x},\boldsymbol{y})$ measures the discrepancy between the linear prediction $\boldsymbol{\theta}^\top\boldsymbol{x}$ and the ground truth $\boldsymbol{y}$, $\tau\in\{0,1\}$ indicates whether there is a regularization term $\frac{\mu}{2}\|\boldsymbol{\theta}\|_2^2$ which aims to enhance generalization ability of the linear model, $d$ denotes the parameter dimension. For the finite-sum problem, each individual loss $\ell(\boldsymbol{\theta}^\top\boldsymbol{x}_i,\boldsymbol{y}_i)$ is associated with the $i$-th sample, while in online setting, the stochastic component $\ell(\boldsymbol{\theta}^\top\boldsymbol{x},\boldsymbol{y};\pi)$ is indexed by a random variable $\pi$. In this work, we assume $F(\boldsymbol{\theta})$ is $\mu$-strongly-convex even in the absence of the regularization $\frac{\mu}{2}\|\boldsymbol{\theta}\|_2^2$. The formulation (1) encapsulates a vast body of important problems, *e.g.* least squares regression, logistic regression and softmax regression, to name a few. In this work, we focus on developing scalable and autonomous first-order optimization methods to solve this fundamental problem which has been extensively studied with a bunch of efficient algorithms, such as gradient descent (GD) [14], stochastic GD (SGD) [15], SDCA [16], SAGA [17], SVRG [18], Catalyst [19], SCSG [20], Catalyst [19], Katyusha [21], and Varag [22].

### 1.1 Motivation

Despite the remarkable success of the stochastic gradient methods and their variance-reduced extensions for solving problem (1), the stochastic gradient evaluation complexity (which usually dominates the computational cost) of these algorithms on the finite-sum problems tends to scale linearly with data size $n$. Such a linear dependence is not only expensive when data scale is huge but also problematic in online and life-long learning regimes where samples are coming infinitely. As pointed out in [20], there are situations in which accurate solutions can be obtained with less than a single pass through the data, *e.g.* for a large-scale dataset with similar and redundant samples. Therefore, developing data-size-independent learning algorithms is of special importance in this big data era.

Particularly, we are interested in efficiently optimizing the finite-sum problem in (1) to its intrinsic excess error bound which typically scales as $\mathcal{O}(1/\sqrt{n})$. As shown in [23], the excess error which measures the expected prediction discrepancy between the optimum model and the learnt model over all possible samples and thus reflects the generalization performance of the model, can be decomposed into model approximation error, estimation error and optimization error. Among them, the model approximation error measures how closely the selected prediction model can approximate the optimal model; the estimation error measures the prediction effects of minimizing the empirical risk instead

- *P. Zhou is with the Sea AI Lab of Sea group, Singapore (email: panzhou3@gmail.com). Part work of this manuscript is done in Salesforce and remaining work is done in Sea.*
- *S. C.H. Hoi is with the Salesforce Research, Singapore (email: shoi@salesforce.com).*
- *X.-T. Yuan (🖂) is with the School of Computer & Software at Nanjing University of Information Science and Technology, China (corresponding author, email: xtyuan@nuist.edu.cn).*
- *Z. Lin is with the School of EECS, Peking University, China, and also with the Cooperative Medianet Innovation Center, Shanghai Jiaotong University, China (email: zlin@pku.edu.cn).*

TABLE 1: Comparison of IFO complexity for first-order stochastic algorithms on the $\mu$-strongly-convex finite-sum problem (1) with linear prediction structure. The solution $\boldsymbol{\theta}$ with $\epsilon$-optimization-error is measured by sub-optimality $\mathbb{E}\left[F(\boldsymbol{\theta})-F(\boldsymbol{\theta}^*)\right]\leq\epsilon$ with optimum $F(\boldsymbol{\theta}^*)$. $\kappa$ denotes the conventional condition number, and $\widetilde{\kappa}=L/\sigma$ where $L$ and $\sigma$ are respectively the smoothness and strongly-convex parameters of $\ell(\boldsymbol{\theta}^\top\boldsymbol{x},\boldsymbol{y})$ w.r.t. $\boldsymbol{\theta}^\top\boldsymbol{x}$. Here we define a set of constants for quadratic (generic) loss: $\zeta=1$ $(2)$, and $\alpha=0$ $(1)$. These different constants only affect the logarithmic factor $\xi=\log\left(\frac{1}{\epsilon}\right)$. For brevity, we define $\rho=\widetilde{\kappa}^\alpha\kappa\log^{1+\zeta}\left(\frac{1}{\epsilon}\right)+\frac{\widetilde{\kappa}^\alpha}{\epsilon}$. The third column (Better Zoom of HSDMPG) summarizes the conditions under which HSDMPG has lower IFO complexity than the compared algorithms. For notations, we define $x\wedge y=\min(x,y)$, $x\vee y=\max(x,y)$. For HSDMPG, it ignores a logarithmic factor $\log d$ in its IFO complexity for brevity, as $\log d$ is often much smaller than $d$, $\kappa$, $n$ and $\frac{1}{\epsilon}$.

| | | $\epsilon$-Optimization Error for ERM (1) | | $\frac{1}{\sqrt{n}}$-Optimization Error for ERM (1) |
|---|---|---|---|---|
| | | IFO Complexity | Better Zoom of HSDMPG | |
| SGD | | $\mathcal{O}\left(\frac{1}{\mu\epsilon}\right)$ | ① $\mu\leq\widetilde{\kappa}^\alpha,\mu\widetilde{\kappa}^\alpha\kappa\epsilon\xi^{\zeta+1}\leq\mathcal{O}(1)$<br>or ② $\mathcal{O}(n)\leq\frac{1}{\mu\epsilon\widetilde{\kappa}^\alpha\xi^\zeta},\mu\widetilde{\kappa}^\alpha\kappa\epsilon\xi^{\zeta+1}\leq\mathcal{O}(1)$ | $\mathcal{O}\left(n\right)$ |
| SVRG, SAGA, APSDCA | | $\mathcal{O}\left((n+\kappa)\log\left(\frac{1}{\epsilon}\right)\right)$ | ① $\rho\xi^{-1}\leq\mathcal{O}\left(n\right)$ | $\mathcal{O}\left(n\log(n)\right)$ |
| APCG | | $\mathcal{O}\left(\frac{n}{\sqrt{\mu}}\log\left(\frac{1}{\epsilon}\right)\right)$ | ① $\mu^{0.5}\rho\xi^{-1}\leq\mathcal{O}\left(n\right)$<br>or ② $\mu\leq\widetilde{\kappa}^\alpha\xi^{1-\zeta},\mu^{0.5}\widetilde{\kappa}^\alpha\kappa\xi^{-\zeta}\leq\mathcal{O}(n)$ | $\mathcal{O}\left(n^{1.25}\log(n)\right)$ |
| SPDC, Catalyst, Katyusha | | $\mathcal{O}\left((n+\sqrt{n\kappa})\log\left(\frac{1}{\epsilon}\right)\right)$ | ① $\rho\xi^{-1}\wedge\rho^2\xi^{-2}\kappa^{-1}\leq\mathcal{O}\left(n\right)$ | $\mathcal{O}\left(n\log(n)\right)$ |
| AMSVRG | | $\mathcal{O}\left(\left(n+\frac{n\kappa}{n+\sqrt{\kappa}}\right)\log\left(\frac{1}{\epsilon}\right)\right)$ | ① $\rho\xi^{-1}\leq\mathcal{O}\left(n\right)$ | $\mathcal{O}\left(n\log(n)\right)$ |
| Varag | | $\mathcal{O}\Big(n\log\left(n\wedge\frac{1}{\epsilon}\right)+\sqrt{n}\Big(\frac{1}{\epsilon^{0.5}}\wedge\kappa^{0.5}\log\left(\frac{1}{\epsilon\kappa}\right)\Big)\Big)$ | ① $\rho\log^{-1}(n\wedge\frac{1}{\epsilon})\leq\mathcal{O}\left(n\right)$<br>or ② $\rho^2(\epsilon\vee\frac{1}{\kappa\log^2(\frac{1}{\epsilon\kappa})})\leq\mathcal{O}\left(n\right)$ | $\mathcal{O}\left(n\log(n)\right)$ |
| SCSG | | $\mathcal{O}\left((n\wedge\frac{\kappa}{\epsilon}+\kappa)\log\left(\frac{1}{\epsilon}\right)\right)$ | ① $\widetilde{\kappa}^\alpha\xi^\alpha\leq\mathcal{O}\left(\kappa\right)$ | $\mathcal{O}\left(n\log(n)\right)$ |
| averaged accelerated SGD (quadratic) | | $\mathcal{O}\left(\frac{Ld}{\epsilon}\right)$ | ① $\frac{\epsilon\xi^2}{\mu}\leq\mathcal{O}\left(d\right)$ | $\mathcal{O}\left(n^{0.5}d\right)$ |
| averaged SGD | quadratic | $\mathcal{O}\left(\frac{\sqrt{L}+\sqrt{d}}{\epsilon}\right)$ | ① $\kappa^2\epsilon^2\xi^4\leq\mathcal{O}(d)$ | $\mathcal{O}\left(n^{0.5}(d^{0.5}+L^{0.5})\right)$ |
| | generic | $\mathcal{O}\left(\frac{\widetilde{\kappa}^3 d}{\epsilon}\right)$ | ① $\frac{\kappa\epsilon\xi^3}{\widetilde{\kappa}^2}\leq\mathcal{O}(d)$ | $\mathcal{O}\left(n^{0.5}d\widetilde{\kappa}^3\right)$ |
| HSDMPG | quadratic | $\mathcal{O}\left(\kappa\log^2\left(\frac{1}{\epsilon}\right)+\frac{1}{\epsilon}\bigwedge n\log\left(\frac{1}{\epsilon}\right)\right)$ | ——————————— | $\mathcal{O}\left(n^{0.5}\log^2(n)\right)$ |
| | generic | $\mathcal{O}\left(\widetilde{\kappa}\kappa\log^3\left(\frac{1}{\epsilon}\right)+\frac{\widetilde{\kappa}}{\epsilon}\bigwedge\widetilde{\kappa}n\log^2\left(\frac{1}{\epsilon}\right)\right)$ | ——————————— | $\mathcal{O}\left(n^{0.5}\widetilde{\kappa}\log^3(n)\right)$ |

of the population risk; and the optimization error denotes the prediction difference between the exact and approximate solutions of empirical risk minimization (ERM). See more details in Sec. 2. Therefore, to achieve small excess error, one should minimize these three kinds of errors jointly. With optimal choice $\mu=\mathcal{O}(1/\sqrt{n})$ to balance empirical risk and generalization gap (namely, the difference in performance of the model on population versus empirical data), the estimation error is known to be of the order $\mathcal{O}(1/\sqrt{n})$, which implies the excess error is at least of the order $\mathcal{O}(1/\sqrt{n})$ [24]–[26]. Thus, it is sufficient to optimize problem (1) to the optimization error $\mathcal{O}(1/\sqrt{n})$ to match the optimal excess error without redundant computation.

## 1.2 Overview of our algorithm and results

The main contribution of this paper is a novel Hybrid Stochastic-Deterministic Minibatch Proximal Gradient (HSDMPG) algorithm with substantially improved complexity over existing methods. Moreover, for large-scale problems, HSDMPG enjoys data-size-independent complexity and thus is scalable. For quadratic problems under both the finite-sum and online settings, the core idea of our method is to recurrently convert the original large-scale ERM problem (1) into a series of minibatch proximal ERM subproblems for efficient minimization and update. Specifically, as a starting point, we uniformly randomly select a minibach $\mathcal{S}$ of components of the risk function $F$ to form a stochastic approximation $F_{\mathcal{S}}$ that will be fixed throughout the algorithm iteration. Next, at each iteration step, we first construct a stochastic surrogate of $F$ by combining the Bregman divergence of $F_{\mathcal{S}}$ at the current iterate and a first-order hybrid stochastic-deterministic approximation of $F$; and then we invoke existing variance-reduced algorithms, such as SVRG, to minimize this surrogate subproblem to desired optimization error. For quadratic loss, we can provably establish sharper bounds of incremental first order oracle (IFO, see Definition 2 in Sec. 3.2) for such a hybrid stochastic-deterministic minibatch proximal update procedure in large-scale settings. To extend the strong efficiency guarantee to generic strongly convex losses, we propose to iteratively convert the non-quadratic problem into a sequence of quadratic subproblems such that the aforementioned method can be readily applied to optimization. In this way, up to logarithmic factors, HSDMPG still enjoys an identical sharp bound of IFO for strongly convex optimization problems.

**Finite-sum Setting.** For finite-sum problems, Table 1 summarizes the computational complexity (measured by IFO) of HSDMPG and several representative baselines, including SGD [15], [27], SVRG [18], SAGA [17], APSDCA [28], APCG [29], SPDC [30], Catalyst [19], Varag [22], AMSVRG [31], Katyusha [21], and SCSG [20], averaged accelerated SGD [32], [33]. We highlight the advantages of our method over these prior approaches below:

1) To achieve $\epsilon$-optimization-error, *i.e.* $\mathbb{E}[F(\boldsymbol{\theta})-F(\boldsymbol{\theta}^*)]\leq\epsilon$, the IFO complexity of HSDMPG on the finite-sum problem in (1) is $\mathcal{O}\Big(\widetilde{\kappa}^\alpha\kappa\log^{\zeta+1}\left(\frac{1}{\epsilon}\right)+\frac{\widetilde{\kappa}^\alpha}{\epsilon}\bigwedge\widetilde{\kappa}^\alpha n\log^\zeta\left(\frac{1}{\epsilon}\right)\Big)$ where $\zeta=1$, $\alpha=0$ for quadratic loss and $\zeta=2$, $\alpha=1$ for generic strongly convex loss. In comparison, the IFO complexity bounds of all the compared algorithms except (averaged accelerated) SGD and SCSG scale linearly w.r.t. the data size $n$. As specified in the third column (Better Zoom of HSDMPG) of Table 1, HSDMPG is superior to these algorithms in large-scale problem settings which are of central interest in big data applications. Compared with SGD, as in most cases the condition number $\kappa$ is of the order $\mathcal{O}(1/\mu)$, HSDMPG improves over SGD by a factor at least $\mathcal{O}\big(\frac{\kappa}{\widetilde{\kappa}^\alpha}\wedge\frac{1}{\widetilde{\kappa}^\alpha\epsilon}\big)$ (up to logarithmic factors). For SCSG, if we ignore the logarithmic factors, our HSDMPG improves the factor $\mathcal{O}\left(\frac{\kappa}{\epsilon}\right)$ in SCSG to $\mathcal{O}\left(\frac{\widetilde{\kappa}^\alpha}{\epsilon}\right)$. So on quadratic problems, HSDMPG strictly improves SGD and SCSG, since $\widetilde{\kappa}^\alpha=1$. For generic problems, when $\kappa\geq\widetilde{\kappa}$ which often holds in practice as the commonly used data $\{\boldsymbol{x}_i\}_{i=1}^n$ has bounded norm, HSDMPG also enjoys lower computational complexity than both SGD and SCSG. See more detailed discussion in Sec. 4.2.

Finally, Table 1 shows that if the problems are of high-dimension or the optimization error $\epsilon$ is very small, then HSDMPG has lower complexity than averaged (accelerated) SGD.

2) For the practical setting where $\epsilon = \mathcal{O}(1/\sqrt{n})$ which matches the optimal intrinsic excess error, HSDMPG has the IFO complexity $\mathcal{O}\left(n^{0.5}\log^2(n)\right)$ for the quadratic loss and $\mathcal{O}\left(n^{0.5}\widetilde{\kappa}^{\alpha}\log^3(n)\right)$ for the generic strongly convex loss. By ignoring the small logarithmic term $\log(n)$, both complexities of HSDMPG are lower than the complexity bound $\mathcal{O}\big(n\big)$ of SGD by a factor $\mathcal{O}\big(n^{0.5}/\widetilde{\kappa}^{\alpha}\big)$. Similarly, HSDMPG improves over APCG and other remaining algorithms except for averaged (accelerated) SGD, by factors of $\mathcal{O}\big(n^{0.75}/\widetilde{\kappa}^{\alpha}\big)$ and $\mathcal{O}\big(n^{0.5}/\widetilde{\kappa}^{\alpha}\big)$, respectively. Moreover, HSDMPG also outperforms averaged (accelerated) SGD on high dimensional problems. These results demonstrate the superior computational efficiency of HSDMPG for attaining near-optimal generalization rate of a statistical learning model.

**Online Setting.** For the online version of problem (1), we establish the IFO complexity bound $\mathcal{O}\left(\widetilde{\kappa}^{\alpha}\kappa\log^{\zeta+1}\left(\frac{1}{\epsilon}\right)+\frac{\widetilde{\kappa}^{\alpha}}{\epsilon}\right)$ where $\zeta = 1, \alpha = 0$ for quadratic problems and $\zeta = 2, \alpha = 1$ for generic problems. Under this setting, SGD and SCSG have IFO complexity $\mathcal{O}\big(\frac{1}{\mu\epsilon}\big)$ and $\mathcal{O}\left(\frac{\kappa}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)$ respectively, while other algorithms in Table 1 are not applicable or are not equipped with rigorous analysis of convergence and computational complexity. Similar to finite-sum setting, HSDMPG is also faster than SGD by a factor of $\mathcal{O}\big(\frac{\kappa}{\widetilde{\kappa}^{\alpha}}\wedge\frac{1}{\widetilde{\kappa}^{\alpha}\epsilon}\big)$, and enjoys lower computational complexity than SCSG when $\kappa \geq \widetilde{\kappa}^{\alpha}$.

**Discussion with our conference work.** This paper is an extension of our previous work [34] which proposes HSDMPG and analyzes its computational complexity for solving the regularized finite-sum problem in (1). Compared with its short version, this paper makes the following changes. **1)** It slightly modifies the HSDMPG algorithm in [34] to use more flexible Bregman divergence, and accordingly develops more advanced analysis technique which improves the computational complexity in [34]. Specifically, it improves the complexity $\mathcal{O}\big(\widetilde{\kappa}^{\alpha}\kappa\sqrt{s\log(d)}\log^{\zeta+1}\big(\frac{1}{\epsilon}\big)+\big(1+\frac{\kappa^3\log^{1.5}(d)}{s^{1.5}}\big)\frac{\widetilde{\kappa}^{\alpha}}{\epsilon}\bigwedge\big(1+\frac{\kappa\log^{0.5}(d)}{s^{0.5}}\big)\widetilde{\kappa}^{\alpha'}n\log^{\zeta}\big(\frac{1}{\epsilon}\big)\big)$ of HSDMPG in [34] to $\mathcal{O}\left(\widetilde{\kappa}^{\alpha}\kappa\log^{\zeta+1}\big(\frac{1}{\epsilon}\big)+\frac{\widetilde{\kappa}^{\alpha}}{\epsilon}\bigwedge\widetilde{\kappa}^{\alpha}n\log^{\zeta}\big(\frac{1}{\epsilon}\big)\right)$ where $\zeta=1, \alpha=\alpha'=0$ for quadratic problems and $\zeta=2, \alpha=1$, $\alpha' = 3$ for generic problems, where $s > 1$ is the size of the minibatch $\mathcal{S}$ of a stochastic approximation $F_{\mathcal{S}}$ to the risk function $F$ (see details in Sec. 3.1), and $d$ denotes the problem dimension. For the practical setting where $\epsilon = \mathcal{O}(1/\sqrt{n})$ which matches the optimal intrinsic excess error, our current IFO complexity is $\mathcal{O}\big(\widetilde{\kappa}^{\alpha}n^{0.5}\log^{1+\zeta}\big(n\big)\big)$ where $\zeta = 1, \alpha = 0$ for the quadratic loss and $\zeta = 2, \alpha = 1$ for the generic loss, which improves previous ones $\mathcal{O}\big(\widetilde{\kappa}^{\alpha}n^{0.875}\log^{\beta}\big(n\big)\big)$ where $\beta=1.5, \alpha=0$ for the quadratic loss and $\beta=2.25, \alpha=1$ for the generic loss. See more discussion of the algorithm modification and improved theoretical technique in Sec. 3.2.1. **2)** Our previous work [34] only analyzes the regularized finite-sum problems in (1), while this work extends HSDMPG to the regularized and non-regularized finite-sum and online problems, which greatly improves the applicability of HSDMPG. **3)** Experimental results of our modified HSDMPG on the regularized and non-regularized finite-sum and online problems are provided to better demonstrate its computational efficiency.

## 2 Related Work

**Stochastic gradient algorithms.** Gradient descent (GD) [14] method has long been applied to solve ERM and enjoys linear convergence rate on strongly convex problems. But it needs to compute full gradient per iteration, leading to huge computation cost on large-scale problems. To improve efficiency, incremental gradient algorithms have been developed via leveraging the finite-sum structure and have witnessed tremendous progress recently. For instance, SGD [15], [35] only evaluates gradient of one (or a minibatch) randomly selected sample at each iteration, which greatly reduces the cost of each iteration and shows more appealing efficiency than GD on large-scale problems [27], [31], [36], [37]. Along this line of research, a variety of variance-reduced variants, such as SVRG [18], SAGA [17], APSDCA [27], AMSVRG [31], SCSG [20], Catalyst [19], Katyusha [21], and Varag [22], are developed and have delivered exciting progress such as linear convergence rates on strongly convex problems as opposed to sublinear rates of vanilla SGD [27]. The hybrid stochastic-deterministic gradient descent method [38]–[42] iteratively samples an evolving minibatch of samples for gradient estimation or subproblem construction and works favorably in reducing the computational complexity. Our HSDMPG method differs significantly from these prior algorithms. Based on the Bregman-divergence of the minibatch function and a hybrid stochastic-deterministic first-order approximation of the original function, HSDMPG constructs a variance-reduced minibatch proximal function which is provably more efficient. Moreover, HSDMPG can employ any off-the-shelf algorithm to solve the constructed sub-problems in the inner loop and thus is flexible for implementation. HSDMPG shares a similar spirit with the DANE method [43] which also uses a local Bregman-divergence-based function approximation for communication-efficient distributed quadratic loss optimization. One main difference lies in the way of constructing first-order approximation of the risk function: HSDMPG employs a novel hybrid stochastic-deterministic approximation strategy which is substantially more efficient than the deterministic strategy as used by DANE. Moreover, DANE focuses on distributed quadratic problems, while HSDMPG is applicable not only to quadratic problems but also to generic strongly convex problems.

**Generalization and optimization.** In the seminal work of [23], it has been demonstrated that the excess error that measures the generalization performance of an ERM model over a function class can be decomposed into three terms in expectation: an *approximation error* that measures how accurate the function class can approximate the underlying optimum model; an *estimation error* that measures the effects of minimizing ERM instead of population risk; and an *optimization error* that represents the difference between the exact solution and the approximate solution of ERM. Particularly, for the $\ell_2$-regularized convex ERM with linear models as in (1), its estimation error (or excess risk) has long been studied with a vast body of deep theoretical results established [26], [32], [33], [44]–[46]. A simple yet powerful tool for analyzing estimation error is the *stability* of an estimator to the changes of training dataset [47]. Among others, *uniform stability* is one of the most popular and useful notion of stability, which measures the prediction discrepancy of an algorithm/model respectively trained on a vanilla dataset $\mathcal{D}$ and the dataset that includes all samples in $\mathcal{D}$ but randomly removes an arbitrary sample in $\mathcal{D}$. Under this notion, Bousquet et al. [47] showed that the $\ell_2$-regularized convex ERM has uniform stability of the order

$\beta = \mathcal{O}(1/(\mu n))$. On the other hand, the estimation error has been shown to be of the order $\mathcal{O}(\beta + 1/\sqrt{n})$, and the optimal uniform stability $\beta$ should be of the order $\mathcal{O}(1/\sqrt{n})$ [25], [48]. This gives rise to the optimal choice $\mu = \mathcal{O}(1/\sqrt{n})$ for balancing empirical loss and generalization gap to achieve estimation error $\mathcal{O}(1/\sqrt{n})$. It implies that the overall excess error is dominated by $\mathcal{O}(1/\sqrt{n})$. In this sense, it suffices to solve the $\ell_2$-regularized ERM to optimization error $\mathcal{O}(1/\sqrt{n})$ to match the intrinsic excess error.

## 3 HSDMPG Algorithm for Quadratic Loss

In this section, we first introduce the hybrid stochastic-deterministic minibatch proximal gradient (HSDMPG) algorithm for quadratic loss function which actually serves as a basis for optimizing generic strongly convex losses. Then we theoretically analyze its convergence rate and computational complexity for finite-sum and online problems. We extend HSDMPG and its theoretical analysis to generic strongly convex loss functions in Sec. 4.

### 3.1 Algorithm

The HSDMPG method is outlined in Algorithm 1. The initial step is to randomly sample a minibatch $\mathcal{S}$ of data points of size $s$ to construct a stochastic approximation

$$F_{\mathcal{S}}(\boldsymbol{\theta}) = \frac{1}{s}\sum\nolimits_{i\in\mathcal{S}} \ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i) + \frac{\tau\mu}{2}\|\boldsymbol{\theta}\|_2^2 \tag{2}$$

to the original risk function $F(\boldsymbol{\theta})$ in problem (1). $F_{\mathcal{S}}(\boldsymbol{\theta})$ will be fixed throughout the computational procedure to follow. Then in the iteration loop the algorithm iterates between two steps of S1 and S2. In step S1, we uniformly randomly sample a size increasing minibatch $\mathcal{S}_t$ of samples to estimate an inexact function

$$F_{\mathcal{S}_t}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{S}_t|}\sum\nolimits_{i\in\mathcal{S}_t} \ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i) + \frac{\tau\mu}{2}\|\boldsymbol{\theta}\|_2^2. \tag{3}$$

Let $\mathcal{D}_g(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2) = g(\boldsymbol{\theta}_1) - g(\boldsymbol{\theta}_2) - \langle \nabla g(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\rangle$ denote the Bregman divergence of a function $g$. Based on $F_{\mathcal{S}}(\boldsymbol{\theta})$ and $F_{\mathcal{S}_t}(\boldsymbol{\theta})$, we construct a variance-reduced minibatch proximal objective $\widetilde{P}_{t-1}(\boldsymbol{\theta})$ to approximate the objective $F(\boldsymbol{\theta})$ in (1), where

$$\widetilde{P}_{t-1}(\boldsymbol{\theta}) \triangleq F_{\mathcal{S}_t}(\boldsymbol{\theta}_{t-1}) + \langle \nabla F_{\mathcal{S}_t}(\boldsymbol{\theta}_{t-1}), \boldsymbol{\theta} - \boldsymbol{\theta}_{t-1}\rangle + \eta \mathcal{D}_{\widetilde{F}_{\mathcal{S}}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{t-1}). \tag{4}$$

Here $\mathcal{D}_{\widetilde{F}_{\mathcal{S}}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{t-1})$ is the Bregman divergence of a regularized loss $\widetilde{F}_{\mathcal{S}}(\boldsymbol{\theta}) = F_{\mathcal{S}}(\boldsymbol{\theta}) + \frac{\gamma}{2}\|\boldsymbol{\theta}\|_2^2$ which essentially measures the distance between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_{t-1}$ on the current geometry curve estimated on $\widetilde{F}_{\mathcal{S}}(\boldsymbol{\theta})$. There are three reasons that we choose $\widetilde{F}_{\mathcal{S}}(\boldsymbol{\theta}) = F_{\mathcal{S}}(\boldsymbol{\theta}) + \frac{\gamma}{2}\|\boldsymbol{\theta}\|_2^2$ to construct the Bregman divergence $\mathcal{D}_{\widetilde{F}_{\mathcal{S}}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{t-1})$. Firstly, $F_{\mathcal{S}}(\boldsymbol{\theta})$ in $\widetilde{F}_{\mathcal{S}}(\boldsymbol{\theta})$ is an unbiased estimation of the vanilla function $F(\boldsymbol{\theta})$, and can well estimate the local geometry of $F(\boldsymbol{\theta})$. In contrast, other commonly used functions $\widetilde{F}_{\mathcal{S}}$ in $\mathcal{D}_{\widetilde{F}_{\mathcal{S}}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{t-1})$, *e.g.* $\widetilde{F}_{\mathcal{S}} = \frac{1}{2}\|\boldsymbol{\theta}\|_2^2$, usually cannot well estimate the geometry curve of vanilla function $\boldsymbol{F}(\boldsymbol{\theta})$. Secondly, as shown below, $\widetilde{F}_{\mathcal{S}}(\boldsymbol{\theta})$ has finite-sum structure and allows us to use efficient stochastic algorithm to optimize $\widetilde{P}_{t-1}(\boldsymbol{\theta})$, while the commonly used $\widetilde{F}_{\mathcal{S}} = \frac{1}{2}\|\boldsymbol{\theta}\|_2^2$ has not the finite-sum structure and only uses gradient descent to optimize. Thirdly, the regularization $\frac{\gamma}{2}\|\boldsymbol{\theta}\|_2^2$ in $\widetilde{F}_{\mathcal{S}}(\boldsymbol{\theta})$ enhances the strongly-convex parameter of $\widetilde{P}_{t-1}(\boldsymbol{\theta})$ and $\widetilde{P}_{t-1}(\boldsymbol{\theta})$ from $\eta\mu$ to $\eta(\mu+\gamma)$, which benefits the optimization of $\widetilde{P}_{t-1}(\boldsymbol{\theta})$. We define the next iterate as

$$\boldsymbol{\theta}_t = \arg\min\nolimits_{\boldsymbol{\theta}} \widetilde{P}_{t-1}(\boldsymbol{\theta}) = \arg\min\nolimits_{\boldsymbol{\theta}} P_{t-1}(\boldsymbol{\theta}), \tag{5}$$

**Algorithm 1** Hybrid Stochastic-Deterministic Minibatch Proximal Gradient (HSDMPG) for quadratic loss.

```
Input: initialization $\boldsymbol{\theta}_0$, regularization constant $\gamma$ in (5), optimization error $\varepsilon_t$.
Initialization: Uniformly randomly sample a data batch $\mathcal{S}$ of size $s$ to form $F_{\mathcal{S}}(\boldsymbol{\theta})$ in (2).
for $t = 1, 2, \ldots, T$ do
    (S1) Uniformly randomly sample a minibatch $\mathcal{S}_t$ to form $F_{\mathcal{S}_t}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{S}_t|}\sum_{i\in\mathcal{S}_t}\ell(\boldsymbol{\theta}^\top\boldsymbol{x}_i, \boldsymbol{y}_i) + \frac{\tau\mu}{2}\|\boldsymbol{\theta}\|_2^2$ and compute $\nabla F_{\mathcal{S}_t}(\boldsymbol{\theta}_{t-1})$ to construct loss $P_{t-1}(\boldsymbol{\theta})$ in (5).
    (S2) Optimize the subproblem (5), e.g. via SVRG, to obtain $\boldsymbol{\theta}_t$ that satisfies $\mathbb{E}[\|\nabla P_{t-1}(\boldsymbol{\theta}_t)\|_2] \le \varepsilon_t$.
end for
Output: $\boldsymbol{\theta}_T$.
```

where $P_{t-1}(\boldsymbol{\theta}) \triangleq$

$$\langle \nabla F_{\mathcal{S}_t}(\boldsymbol{\theta}_{t-1}), \boldsymbol{\theta}\rangle + \eta\Big[F_{\mathcal{S}}(\boldsymbol{\theta}) - \langle \nabla F_{\mathcal{S}}(\boldsymbol{\theta}_{t-1}), \boldsymbol{\theta}\rangle + \frac{\gamma}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1}\|_2^2\Big].$$

In $P_{t-1}$, its finite-sum structure comes from the initial stochastic approximation $F_{\mathcal{S}}(\boldsymbol{\theta})$ and its gradient at $\boldsymbol{\theta}_{t-1}$. Since along with more iterations, the size of $\mathcal{S}_t$ increases which indicates that the loss $P_{t-1}$ is a variance-reduced loss and will converge to the original loss $F(\boldsymbol{\theta})$ in problem (1). Then in step S2, we approximately solve problem (5) via a stochastic gradient optimization method such as SVRG. The principle behind this strategy is that for the initial optimization progress, inexact gradient already can well decrease the loss since the current solution is far from the optimum, while along more iterations, the current solution becomes closer to optimum, requiring more accurate gradient for further reducing the loss function. In this way, our proposed method can well balance the convergence speed and the computational cost at each iteration and thus has the potential to achieve improved overall computational efficiency. Shamir *et al.* [43] proposed the DANE method which uses a similar local Bregman divergence based regularization for distributed quadratic optimization problems. Our method improves upon DANE in two aspects: 1) we use the variance-reduction techniques to reduce the overall computational complexity, and 2) HSDMPG is applicable not only to quadratic problems but also to generic strongly convex problems with about the same computational complexity which will be discussed in Sec. 4.

### 3.2 Convergence and complexity analysis

For analysis, we first introduce two necessary definitions, *i.e.* strong convexity and Lipschitz smoothness, which are conventionally used in the analysis of convex optimization methods [18], [27].

**Definition 1** (Strong Convexity and Smoothness)**.** *A differentiable function $g(\boldsymbol{\theta})$ is said to be $\mu$-strongly-convex and $L$-smooth if $\forall \boldsymbol{\theta}_1, \boldsymbol{\theta}_2$, it satisfies*

$$\frac{\mu}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2 \le \mathcal{D}_g(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \le \frac{L}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2.$$

*where $\mathcal{D}_g(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2) = g(\boldsymbol{\theta}_1) - g(\boldsymbol{\theta}_2) - \langle \nabla g(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\rangle$.*

For brevity, let $\ell_i(\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i) + \frac{\tau\mu}{2}\|\boldsymbol{\theta}\|_2^2$. Following [18], [30], we employ the incremental first order oracle (IFO) complexity as the computation complexity metric for solving problem (1).

**Definition 2.** *An IFO takes an index $i \in [n]$ and a point $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, and returns the pair $(\ell_i(\boldsymbol{\theta}), \nabla \ell_i(\boldsymbol{\theta}))$.*

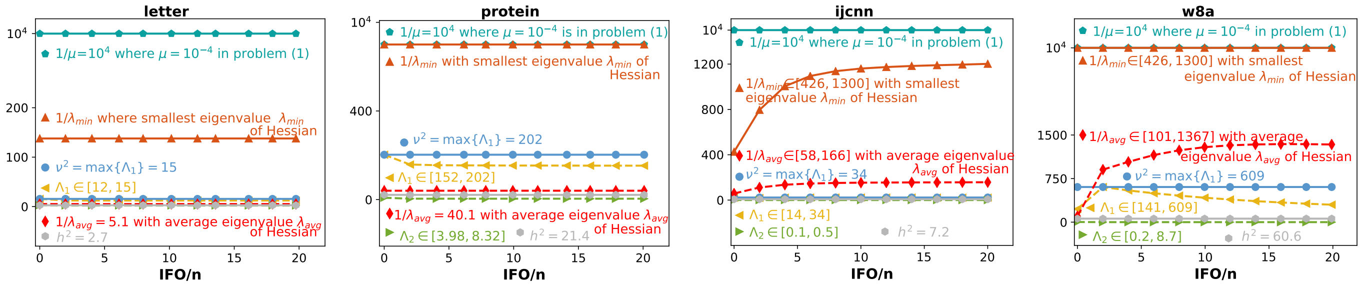


Fig. 1: Investigation of $\nu$ and $h$: stochastic gradient algorithms process data multiple pass on quadratic problems (letter and protein) and logistic regression problems (ijcnn and w8a) where their regularization constant $\mu$ is $\mu=10^{-4}$. We define $\Lambda_1=\frac{1}{n}\sum_{i=1}^n\|\boldsymbol{H}^{-1/2}(\nabla F(\boldsymbol{\theta})-\nabla\ell_i(\boldsymbol{\theta})-\tau\mu\boldsymbol{\theta})\|_2^2$, $\Lambda_2=\frac{1}{n}\sum_{i=1}^n\|\nabla F(\boldsymbol{\theta})-\nabla\ell_i(\boldsymbol{\theta})-\tau\mu\boldsymbol{\theta}\|_2^2$, the smallest eigenvalue $\lambda_{\text{min}}$ and average eigenvalue $\lambda_{\text{avg}}$ of the Hessian $\boldsymbol{H}$ at each iteration. One can observe that $\nu^2(=\max\{\Lambda_1\})$ is often of the order $\mathcal{O}(1/\lambda_{\text{avg}})$ and is much smaller than $\mathcal{O}(1/\lambda_{\text{min}})$ and $h^2$ is often at the same order as $\nu^2$. **Best viewed in $\times$2 sized color pdf file**.

The IFO complexity can accurately reflect the overall computational performance of a first-order algorithm, as objective value and gradient evaluation usually dominate the per-iteration complexity. Next, we provide convergence analysis of HSDMPG on finite-sum and online quadratic problems in Sec. 3.2.1 and Sec. 3.2.2, respectively. In the following analysis, for brevity, let $\boldsymbol{H}$ be the Hessian matrix of the quadratic function $F(\boldsymbol{\theta})$ and $\|\boldsymbol{\theta}\|_{\boldsymbol{H}}=\sqrt{\boldsymbol{\theta}^\top\boldsymbol{H}\boldsymbol{\theta}}$. We always suppose that $\|\boldsymbol{x}_i\|\leq r,\forall i$, which generally holds for natural data analysis, *e.g.*, in computer vision and signal processing.

### 3.2.1 Finite-sum setting

Now we analyze HSDMPG under finite-sum setting. We summarize our main result in Theorem 1 which shows the linear convergence rate of HSDMPG for quadratic problems.

**Theorem 1.** *Assume each loss $\ell(\boldsymbol{\theta}^\top\boldsymbol{x}_i,\boldsymbol{y}_i)$ is quadratic, and $\sup_{\boldsymbol{\theta}\in\Theta}\frac{1}{n}\sum_{i=1}^n\|\boldsymbol{H}^{-1/2}(\nabla F(\boldsymbol{\theta})-\nabla\ell_i(\boldsymbol{\theta})-\tau\mu\boldsymbol{\theta})\|_2^2\leq\nu^2$, where the set $\Theta$ contains the sequence $\{\boldsymbol{\theta}_t\}_{t=0}^T$ produced by Algorithm 1. Consider the following two cases:*
*(1) when the empirical risk $\frac{1}{n}\sum_{i=1}^n\ell(\boldsymbol{\theta}^\top\boldsymbol{x}_i,\boldsymbol{y}_i)$ is $\mu$-strongly convex, we do not impose any regularization, namely $\tau=0$, in (1);*
*(2) when the empirical risk $\frac{1}{n}\sum_{i=1}^n\ell(\boldsymbol{\theta}^\top\boldsymbol{x}_i,\boldsymbol{y}_i)$ is only convex, we impose the regularization $\frac{\mu}{2}\|\boldsymbol{\theta}\|_2^2$ where $\tau=1$ in (1).*
*For both cases, we set $\eta=2$, $\varepsilon_t=\frac{\mu^{1.5}}{4(3\mu+4\gamma)}\exp\big(-\frac{\mu(t-1)}{2(3\mu+4\gamma)}\big)$, $\gamma=\frac{1}{2}\Big(\frac{28r^2}{3s}\log\Big(\frac{2d(T+1)}{\delta}\Big)-\mu\Big)^+$, $|\mathcal{S}_t|=\frac{16\nu^2(3\mu+4\gamma)^2}{\mu^2}\exp\Big(\frac{\mu t}{3\mu+4\gamma}\Big)\bigwedge n$, where $d$ is the problem dimension, $x^+$ denotes $\max(x,0)$. Then if $s\geq\frac{28}{3}\log\Big(\frac{2d(T+1)}{\delta}\Big)\wedge n$, with probability at least $1-\delta$ the sequence $\{\boldsymbol{\theta}_t\}$ produced by Algorithm 1 for cases (1) and (2) obeys*

$$\mathbb{E}[F(\boldsymbol{\theta}_t)-F(\boldsymbol{\theta}^*)]=\tfrac{1}{2}\mathbb{E}[\|\boldsymbol{\theta}_t-\boldsymbol{\theta}^*\|_{\boldsymbol{H}}^2]\leq\zeta\exp\Big(-\frac{\mu t}{3\mu+4\gamma}\Big),$$

*where $\zeta=\frac{1}{2}\left(\|\boldsymbol{\theta}_0-\boldsymbol{\theta}_*\|_{\boldsymbol{H}}+\frac{1}{9}\right)^2+\frac{3}{8}$, and the expectation is taken on the randomness of sampling minibatch $\mathcal{S}_t$ to construct the inner subproblem (5) and the randomness of SVRG to solve the subproblem (5).*

See its proof in Appendix B.1. The randomness in Theorem 1 comes from two aspects: 1) the initial step which randomly samples a minibatch $\mathcal{S}$ to construct a stochastic approximation $F_{\mathcal{S}}(\boldsymbol{\theta})$ in (2) to the original function $\boldsymbol{F}(\boldsymbol{\theta})$ and leads to the probability $1-\delta$ in Theorem 1; 2) the subsequent random sampling in the algorithm, such as sampling minibatch $\mathcal{S}_t$ for constructing $F_{\mathcal{S}_t}(\boldsymbol{\theta})$ in (3) and sampling minibatch in SVRG to minimize $F_{\mathcal{S}_t}(\boldsymbol{\theta})$, which gives the expectation in Theorem 1. These two sources of randomness are actually independent of each other. More specifically, the randomness on the probability $1-\delta$ comes from Lemma 3 in Appendix, where we bound $\frac{2\mu}{3\mu+4\gamma}\leq\left\|\boldsymbol{H}^{1/2}\left(\boldsymbol{H}_{\mathcal{S}}+\gamma\boldsymbol{I}\right)^{-1}\boldsymbol{H}^{1/2}\right\|\leq 2$ and $\left\|\boldsymbol{I}-\lambda\boldsymbol{H}^{1/2}\left(\boldsymbol{H}_{\mathcal{S}}+\gamma\boldsymbol{I}\right)^{-1}\boldsymbol{H}^{1/2}\right\|\leq\max\left(1-2\lambda,1-\frac{2\mu}{3\mu+4\gamma}\right)$, in which $\boldsymbol{H}$ and $\boldsymbol{H}_{\mathcal{S}}$ respectively denote the Hessian matrix of $F(\boldsymbol{\theta})$ and $F_{\mathcal{S}}(\boldsymbol{\theta})$ in problem (1).

**Algorithm 2** SVRG for solving the inner problem $P_{t-1}(\boldsymbol{\theta})$.

```
Input: initialization $\widetilde{\boldsymbol{\theta}}^0=\boldsymbol{\theta}_{t-1}$ where $\boldsymbol{\theta}_{t-1}$ is the previous approximation solution for $\min_{\boldsymbol{\theta}} P_{t-2}(\boldsymbol{\theta})$, minibatch size $s'$ at each iteration, epoch length $m$, step size $\eta'$, epoch number $Q$, the gradient $\nabla F_{\mathcal{S}_t}(\boldsymbol{\theta}_{t-1})$.
for $q=1,\ldots,Q$ do
  (S1) $\widetilde{\boldsymbol{\theta}}=\widetilde{\boldsymbol{\theta}}_{q-1}$
  (S2) $\widetilde{\boldsymbol{g}}=\nabla P_{t-1}(\boldsymbol{\theta})=\nabla F_{\mathcal{S}_t}(\boldsymbol{\theta}_{t-1})+\eta\big[\nabla F_{\mathcal{S}}(\widetilde{\boldsymbol{\theta}})-\nabla F_{\mathcal{S}}(\boldsymbol{\theta}_{t-1})+\gamma(\widetilde{\boldsymbol{\theta}}-\boldsymbol{\theta}_{t-1})\big]$.
  (S3) $\boldsymbol{\theta}^0=\widetilde{\boldsymbol{\theta}}$
  for $k=1,\ldots,m$ do
    (S4) uniformly randomly sample a minibatch $\mathcal{S}'_k$ from $\mathcal{S}$
    (S5) $\boldsymbol{v}^k=\nabla P_{t-1}^{\mathcal{S}'_k}(\boldsymbol{\theta}^{k-1})-\nabla P_{t-1}^{\mathcal{S}'_k}(\widetilde{\boldsymbol{\theta}})+\widetilde{\boldsymbol{g}}=\eta\big[\nabla F_{\mathcal{S}}^{\mathcal{S}'_k}(\boldsymbol{\theta}^{k-1})-\nabla F_{\mathcal{S}}^{\mathcal{S}'_k}(\widetilde{\boldsymbol{\theta}})+\gamma(\boldsymbol{\theta}^{k-1}-\widetilde{\boldsymbol{\theta}})\big]+\widetilde{\boldsymbol{g}}$, where $F_{\mathcal{S}}^{\mathcal{S}'_k}(\boldsymbol{\theta}^{k-1})=\frac{1}{|\mathcal{S}'_k|}\sum_{i\in\mathcal{S}'_k}\ell((\boldsymbol{\theta}^{k-1})^\top\boldsymbol{x}_i,\boldsymbol{y}_i)+\frac{\tau\mu}{2}\|\boldsymbol{\theta}\|_2^2$
    (S6) $\boldsymbol{\theta}^k=\boldsymbol{\theta}^{k-1}-\eta'\boldsymbol{v}^k$
  end for
  $\widetilde{\boldsymbol{\theta}}_q=\boldsymbol{\theta}^m$
end for
Output: $\boldsymbol{\theta}_t=\widetilde{\boldsymbol{\theta}}_Q$.
```

The main message conveyed by Theorem 1 is that HSDMPG enjoys linear convergence rate on the quadratic loss when we use evolving size of the minibatch $\mathcal{S}_t$. The assumption that $\sup_{\boldsymbol{\theta}}\frac{1}{n}\sum_{i=1}^n\|\boldsymbol{H}^{-1/2}(\nabla F(\boldsymbol{\theta})-\nabla\ell_i(\boldsymbol{\theta})-\tau\mu\boldsymbol{\theta})\|_2^2\leq\nu^2$ in HSDMPG is mild, which requires the variance of stochastic gradient under the Hessian matrix to be bounded. Such an assumption is analogous to the one used in analysis of SGD that imposes the bounded-variance assumption on stochastic gradient, namely, $\frac{1}{n}\sum_{i=1}^n\|\nabla F(\boldsymbol{\theta})-\nabla\ell_i(\boldsymbol{\theta})-\tau\mu\boldsymbol{\theta}\|_2^2$.

In the statement of SVRG Algorithm 2 in the initialization step gradient for $F_{\mathcal{S}}(\theta_{t-1})$ should also be included. Also, check if the output of Algorithm should be $\theta_t$.

Based on this result, we further analyze the computational complexity of HSDMPG to better understand its overall efficiency in computation. At each iteration, we use the SVRG method solve the inner-loop subproblem (5) because it only accesses the first-order information of the objective function and is efficient. The optimization details of SVRG is summarized in Algorithm 2. Then we summarize our main result on the computation complexity of HSDMPG in Corollary 1 with proof provided in Appendix B.2.

**Corollary 1.** *Suppose that the assumptions in Theorem 1 hold, the inner-loop subproblems are solved via SVRG, and* $s \geq \frac{r^2 \log\left(\frac{d}{\delta}\right)}{\mu} \wedge n$. *For both cases (1) and (2) in Theorem 1, to achieve* $\mathbb{E}[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^*)] \leq \epsilon$ *on the quadratic loss, with probability* $1-\delta$ *the IFO complexity of* HSDMPG *is of the order*

$$\mathcal{O}\Big(\big(\kappa + \log(d)\big) \log^2\left(\frac{1}{\epsilon}\right) + \frac{\nu^2}{\epsilon} \bigwedge n \log\left(\frac{1}{\epsilon}\right)\Big),$$

*where* $\kappa = L/\mu$ *denotes the conditional number.*

Since for most problems, the logarithmic factor $\log(d)$ is usually much smaller than other factors, such as $\kappa$, $n$ and $\frac{1}{\epsilon}$, we will ignore it in the comparison of results unless otherwise stated. Compared with those algorithms in Table 1 whose IFO complexity scales linearly with the data size $n$, *e.g.* SVRG, APCG, Katyusha and AMSVRG, the proposed HSDMPG has data-size-independent IFO complexity and can outperform them for large-scale learning problems where the data size $n$ could be huge. To be more precise, the third column of Table 1 summarizes the conditions under which HSDMPG outperforms these algorithms in terms of the computational complexity.

Next, we compare HSDMPG with the algorithms in Table 1 whose IFO complexity does not linearly scale with $n$. For SGD whose IFO complexity is $\mathcal{O}\big(\frac{1}{\mu\epsilon}\big)$, HSDMPG also enjoys substantially lower complexity in most cases. Concretely, since $\nu$ satisfies $\sup_{\boldsymbol{\theta}\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{H}^{-1/2}(\nabla F(\boldsymbol{\theta}) - \nabla \ell_i(\boldsymbol{\theta}) - \tau\mu\boldsymbol{\theta})\|_2^2 \leq \nu^2$, we know $\nu \leq \mathcal{O}\big(\frac{1}{\mu^{0.5}}\big)$ where $\mu$ denotes the strongly convex parameter of the optimization problem. Moreover, the condition number $\kappa$ is typically of the order $\mathcal{O}\big(1/\mu\big)$. Thus, HSDMPG is always at least as good as SGD, and is of higher efficiency than SGD when the optimization error $\epsilon$ is very small to satisfy $\epsilon \leq \mathcal{O}\big(\frac{1}{\mu n}\big)$. Moreover, if the quadratic problems are of high-dimension or the optimization error $\epsilon$ is very small, then HSDMPG has lower complexity than averaged (accelerated) SGD.

For SCSG, its IFO complexity is $\mathcal{O}\left((n \wedge \frac{h^2\kappa}{\epsilon} + \kappa) \log\left(\frac{1}{\epsilon}\right)\right)$, where $h^2 = \frac{1}{n}\sum_{i=1}^{n} \|\nabla \ell_i(\boldsymbol{\theta}^*) + \tau\mu\boldsymbol{\theta}^*\|_2^2$ and $\boldsymbol{\theta}^*$ is the minimizer of $\boldsymbol{F}(\boldsymbol{\theta})$. By ignoring the logarithmic factors $\log\left(\frac{1}{\epsilon}\right)$ and $\log d$ which are usually much smaller than $n$, $\kappa$ and $\frac{1}{\epsilon}$, we only need to compare $h^2\kappa$ in SCSG and $\nu^2$ in our HSDMPG. Let us study these factors in the real problems. Specifically, we consider regularized least square and logistic regression problems and set their regularization constant $\mu = 10^{-4}$ in (1). Figure 1 investigates six parameters at each iteration, including 1) $\Lambda_1 = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{H}^{-1/2}(\nabla F(\boldsymbol{\theta}) - \nabla \ell_i(\boldsymbol{\theta}) - \tau\mu\boldsymbol{\theta})\|_2^2$, 2) $\Lambda_2 = \frac{1}{n}\sum_{i=1}^{n} \|\nabla F(\boldsymbol{\theta}) - \nabla \ell_i(\boldsymbol{\theta}) - \tau\mu\boldsymbol{\theta}\|_2^2$, 3) $\nu^2$ which equals to the largest $\Lambda_1$, 4) the smallest eigenvalue $\lambda_{\text{min}}$ of the Hessian $\boldsymbol{H}$, 5) the average eigenvalue $\lambda_{\text{avg}}$ of $\boldsymbol{H}$; 6) $h^2 = \frac{1}{n}\sum_{i=1}^{n} \|\nabla \ell_i(\boldsymbol{\theta}^*) + \tau\mu\boldsymbol{\theta}^*\|_2^2$. To estimate the optimum $\boldsymbol{\theta}^*$, we run full gradient descent sufficiently long until $\|\nabla F(\tilde{\boldsymbol{\theta}})\|_2 \leq 10^{-10}$ and approximate $\boldsymbol{\theta}^*$ as $\tilde{\boldsymbol{\theta}}$. We first analyze $\nu^2$. By comparison, one can observe that 1) $\Lambda_1$ is much smaller than $1/\lambda_{\text{min}}$; 2) $\Lambda_1$ is usually of the order $\Lambda_2/\lambda_{\text{avg}}$ and is much smaller than $\Lambda_2/\lambda_{\text{min}}$. This means that $(\nabla F(\boldsymbol{\theta}) - \nabla \ell_i(\boldsymbol{\theta}))$ does not well align with the eigenvector direction of the smallest eigenvalue $\lambda_{\text{min}}$, but is relatively independent of the eigenvector directions of Hessian $\boldsymbol{H}$. Since typically, the average eigenvalue $\lambda_{\text{avg}}$ is much larger than the smallest eigenvalue $\lambda_{\text{avg}}$, then $\nu^2 \approx \mathcal{O}(1/\lambda_{\text{avg}})$ is much smaller than the condition number $\kappa = \mathcal{O}(1/\lambda_{\text{min}})$. Therefore, we can view $\nu^2$ as a constant in this work. Then we look at $h^2$. One can observe that $\nu^2$ is at most eleven times larger than $h^2$. This means that $\nu^2$ and $h^2$ are at the same order, and thus can be viewed as constants. In this way, one can conclude that HSDMPG often enjoys lower computational complexity than SCSG on (moderately) ill-conditioned problems in which the condition number $\kappa$ usually satisfies $\kappa \geq \mathcal{O}\big(\frac{\nu^2}{h^2} \log^2\left(\frac{1}{\epsilon}\right)\big)$, namely the conditions in Table 1.

The complexity in Corollary 1 is superior to our previous one $\mathcal{O}\big(\kappa\sqrt{s\log(d)} \log^2\big(\frac{1}{\epsilon}\big) + \big(1 + \frac{\kappa^3 \log^{1.5}(d)}{s^{1.5}}\big)\frac{1}{\epsilon} \bigwedge \big(1 + \frac{\kappa \log^{0.5}(d)}{s^{0.5}}\big) n \log\big(\frac{1}{\epsilon}\big)\big)$ of HSDMPG in [34]. There are two reasons that leads to this improvement. 1) We add one regularization constant $\eta$ to weight the Bregman divergence $\mathcal{D}_{\widetilde{F}_{\mathcal{S}}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{t-1})$ in (4) more flexibly, which is contrast to the fixed $\eta = 1$ in [34]. It allows us to derive better choice of $\eta$ and thus lower complexity. 2) We advance our analysis technique by improving the lower bound of $R = \|\boldsymbol{H}^{1/2}(\boldsymbol{H}_{\mathcal{S}} + \gamma\boldsymbol{I})^{-1}\boldsymbol{H}^{1/2}\|$ from $R \geq \frac{\mu}{\mu+2\gamma}$ with $\gamma = \frac{(\log^{0.5}(d)+\sqrt{2})Lr^2}{s^{0.5}}$ in [34] to $R \geq \frac{3\mu}{3\mu+4\gamma}$ with $\gamma = \frac{1}{2}\Big(\frac{28r^2}{3s}\log\Big(\frac{2d(T+1)}{\delta}\Big) - \mu\Big)^+$, where $x^+$ denotes $\max(x, 0)$. By comparison, one can observe that we improve $\gamma$ from $\mathcal{O}(\frac{1}{s^{0.5}})$ to $\mathcal{O}(\frac{1}{s})$ which can better lower bound $R$. It should be mentioned that $R$ measures the difference between the Hessian $(\boldsymbol{H}_{\mathcal{S}} + \gamma\boldsymbol{I})$ of our stochastic approximation $F_{\mathcal{S}}(\boldsymbol{\theta})$ and the Hessian $\boldsymbol{H}$ of original function $F_{\mathcal{S}}(\boldsymbol{\theta})$, and plays key role in our complexity analysis. See details in Lemma 3 and the proof of Corollary 1 in Appendix.

From a perspective of generalization, we are particularly interested in the computational complexity of HSDMPG for optimizing the ERM model (1) to its intrinsic excess error bound which characterizes the generalization performance of the model. As reviewed in Sec. 2, the excess error of the considered $\ell_2$-ERM model is typically of order $\mathcal{O}(1/\sqrt{n})$. Accordingly, one only needs to solve the optimization problem to the optimization error $\epsilon = \mathcal{O}(1/\sqrt{n})$ [23], [25]. Moreover, to accord with this intrinsic excess error bound, the regularization constant $\mu$ should also be of the order $\mathcal{O}(\frac{1}{\sqrt{n}})$. In this way, the condition number $\kappa$ could scale as large as $\mathcal{O}(\sqrt{n})$. Based on these results and Corollary 1, we can derive the IFO complexity bound of HSDMPG for this case in Corollary 2. See its proof in Appendix B.3.

**Corollary 2.** *Suppose that the assumptions in Corollary 1 hold. For both cases (1) and (2) in Theorem 1, with probability at least* $1-\delta$, *the IFO complexity of* HSDMPG *on the quadratic loss to achieve* $\mathbb{E}[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^*)] \leq \frac{1}{\sqrt{n}}$ *is of order* $\mathcal{O}\big((n^{0.5} + \log(d))\log^2(n) + \nu^2 n^{0.5}\big)$.

From Corollary 2, one can observe that the IFO complexity of HSDMPG for quadratic problems is of the order $\mathcal{O}\left(n^{0.5}\log^2(n)\right)$, where we ignore the constant $\nu^2$ and the logarithmic factor $\log(d)$ since as aforementioned, $\nu^2$ is much smaller than $1/\mu = \mathcal{O}\left(n^{0.5}\right)$, and $\log(d)$ is often much smaller than $n$. It means that HSDMPG can reach the intrinsic excess error $\mathcal{O}\big(1/\sqrt{n}\big)$ with strictly less than a single pass over the entire training dataset. In comparison, we can observe from Table 1 that in the same practical setting, SGD and APCG have IFO complexity $\mathcal{O}\left(n\right)$ and $\mathcal{O}\left(n^{1.25}\log(n)\right)$, respectively. By ignoring the logarithmic factor $\log(n)$ which is much smaller than $n$ for large-scale learning problems, HSDMPG improves over these two methods by factors $\mathcal{O}\big(n^{0.5}\big)$ and $\mathcal{O}\big(n^{0.725}\big)$, respectively. The IFO complexity of all other algorithms in Table 1, including SVRG, SCSG, SPDC, APSDCA, AMSVRG, Catalyst, Katyusha and Varag, are all of the order $\mathcal{O}\left(n\log\left(n\right)\right)$. Similarly, by ignoring the logarithmic factors, HSDMPG has lower IFO complexity than these algorithms by a factor $\mathcal{O}\big(n^{0.5}\big)$. The complexity in Corollary 2 also improves our previous complexity $\mathcal{O}\left(n^{0.875}\log^{1.5}\left(n\right)\right)$ in [34].

To summarize this group of results comparison, HSDMPG would be significantly superior to all these state-of-the-arts when solving quadratic optimization problems up to the intrinsic excess error.

### 3.2.2 *Online setting*

Now we provide the analysis results of HSDMPG for the online setting. Our main results are stated in Theorem 2.

**Theorem 2.** *Assume that each loss $\ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i)$ is quadratic and $L$-smooth w.r.t. $\boldsymbol{\theta}^\top \boldsymbol{x}_i$, and $\sup_{\boldsymbol{\theta}\in\Theta}\mathbb{E}_i[\|\boldsymbol{H}^{-1/2}(\nabla F(\boldsymbol{\theta}) - \nabla \ell_i(\boldsymbol{\theta}) - \tau\mu\boldsymbol{\theta})\|_2^2] \leq \nu^2$, where the set $\Theta$ contains the sequence $\{\boldsymbol{\theta}_t\}_{t=0}^T$ produced by Algorithm 1. Consider the following two cases:*
*(1) when the population risk $\mathbb{E}[\ell(\boldsymbol{\theta}^\top \boldsymbol{x}, \boldsymbol{y}; \pi)]$ is $\mu$-strongly convex, we do not impose any regularization, namely $\tau = 0$ in (1);*
*(2) when the population risk $\mathbb{E}[\ell(\boldsymbol{\theta}^\top \boldsymbol{x}, \boldsymbol{y}; \pi)]$ is only convex, we impose the regularization $\frac{\mu}{2}\|\boldsymbol{\theta}\|_2^2$, where we set $\tau = 1$ in (1).*
*For both cases, we use the same parameter setting in Theorem 1 except $|\mathcal{S}_t| = \frac{16\nu^2(\mu+2\gamma)^2}{\mu^2}\exp\left(\frac{\mu t}{2(\mu+2\gamma)}\right)$. Then for both cases (1) and (2), with probability at least $1-\delta$, the sequence $\{\boldsymbol{\theta}_t\}$ produced by Algorithm 1 satisfies*

$$\mathbb{E}[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^*)] = \tfrac{1}{2}\mathbb{E}[\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_{\boldsymbol{H}}^2] \leq \zeta \exp\left(-\tfrac{\mu t}{\mu+2\gamma}\right),$$

*where $\zeta = \frac{1}{2}\left(\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*\|_{\boldsymbol{H}} + \frac{1}{2}\right)^2 + \frac{5}{8}$, and the expectation is taken on the randomness of sampling minibatch $\mathcal{S}_t$ to construct the inner subproblem (5) and the randomness of SVRG to solve the subproblem (5). Moreover, to achieve $\mathbb{E}[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^*)] \leq \epsilon$, with probability at least $1-\delta$ the IFO complexity is of the order*

$$\mathcal{O}\Big(\big(\kappa + \log(d)\big)\log^2\Big(\frac{1}{\epsilon}\Big) + \frac{\nu^2}{\epsilon}\Big),$$

*where $\kappa = L/\mu$ denotes the conditional number.*

See its proof in Appendix B.4. Similar to Theorem 1, the randomness in Theorem 2 comes from two aspects, including (i) the randomness in randomly sampling a minibatch $\mathcal{S}$ to construct a stochastic approximation $F_{\mathcal{S}}(\boldsymbol{\theta})$, and (ii) the subsequent random sampling in the algorithm, e.g. sampling minibatch $\mathcal{S}_t$ for constructing $F_{\mathcal{S}_t}(\boldsymbol{\theta})$ in (3) and sampling minibatch in SVRG to minimize $F_{\mathcal{S}_t}(\boldsymbol{\theta})$. Please refer to the discussion below Theorem 1. Theorem 2 shows that with almost the same assumptions and parameter settings as finite-sum problems, for online-setting HSDMPG also enjoys the same linear convergence rate in finite-sum setting. For the computational complexity, it is the same as the one for finite-sum problems. These results demonstrate the superior transferability of HSDMPG.

Under online setting, our proposed HSDMPG is consistently more efficient than SGD. Specifically, by ignoring the logarithmic factor, HSDMPG is at least $\mathcal{O}\big(\kappa \wedge \frac{1}{\epsilon}\big)$ times faster than SGD in terms of IFO complexity. The complexity of SCSG becomes $\mathcal{O}\left(\frac{\kappa}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)$ which is higher than ours, since we always have $\kappa \geq \mathcal{O}(\nu^2)$. Actually, as shown in Figure 1, $\nu^2$ is much smaller than $\kappa$ in practice, indicating much higher computational efficiency of HSDMPG over SCSG. For the remaining algorithms in Table 1, they are not applicable or are not equipped with rigorous analysis of convergence and computational complexity for online convex problems. These results also show the advantages of HSDMPG.

# 4 HSDMPG FOR GENERIC CONVEX LOSS

## 4.1 Algorithm

The computational complexity guarantees established in the previous section are only applicable to quadratic loss function. In order to extend these results to non-quadratic convex loss function, we apply a quadratic approximation strategy to convert the original non-quadratic problem into a sequence of quadratic optimization sub-problems such that each of the subproblem can be optimized by HSDMPG. More specifically, suppose that the loss function $\ell(\boldsymbol{\theta}^\top \boldsymbol{x}, \boldsymbol{y})$ is twice differentiable w.r.t. $\boldsymbol{\theta}^\top \boldsymbol{x}$ and is $L$-smooth w.r.t. $\boldsymbol{\theta}^\top \boldsymbol{x}$. Then we can verify that for all $\boldsymbol{\theta}$,

$$\nabla^2 F(\boldsymbol{\theta}) = \frac{1}{n}\sum\nolimits_{i=1}^{n} \ell''(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i)\boldsymbol{x}_i\boldsymbol{x}_i^\top + \mu\boldsymbol{I} \preceq \bar{\boldsymbol{H}},$$

where $\bar{\boldsymbol{H}} \triangleq \frac{L}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^\top + \mu\boldsymbol{I}$. Therefore, at each iteration, we construct an upper bound of the second-order Taylor expansion of $F$ at $\boldsymbol{\theta}_{t-1}$ as expressed by

$$\boldsymbol{Q}_{t-1}(\boldsymbol{\theta}) \triangleq F(\boldsymbol{\theta}_{t-1}) + \langle\nabla F(\boldsymbol{\theta}_{t-1}), \boldsymbol{\theta} - \boldsymbol{\theta}_{t-1}\rangle + \Delta_{t-1}(\boldsymbol{\theta}), \quad (6)$$

where $\Delta_{t-1}(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1})^\top \bar{\boldsymbol{H}}(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1})$. The finite-sum structure in $\boldsymbol{Q}_{t-1}(\boldsymbol{\theta})$ is from $\nabla F(\boldsymbol{\theta}_{t-1}) = \frac{1}{n}\sum_{i=1}^{n}\nabla\ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i) + \mu\boldsymbol{\theta}$ and $\bar{\boldsymbol{H}}$. Thus, we first adopt $\boldsymbol{\theta}_{t-1}$ as a warm-start initialization of HSDMPG, and then apply HSDMPG to the quadratic function $\boldsymbol{Q}_{t-1}(\boldsymbol{\theta})$ for estimating $\boldsymbol{\theta}_t$ such that

$$\boldsymbol{Q}_{t-1}(\boldsymbol{\theta}_t) \leq \min\nolimits_{\boldsymbol{\theta}} \boldsymbol{Q}_{t-1}(\boldsymbol{\theta}) + \varepsilon'_t. \quad (7)$$

The above nested-loop computation procedure is summarized in Algorithm 3. We remark that when computing the gradient of $\boldsymbol{Q}_{t-1}(\boldsymbol{\theta})$, we can compute the gradient associated with $\bar{\boldsymbol{H}}$ at the point $\boldsymbol{\theta}$ as $\bar{\boldsymbol{H}}(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1}) = \frac{L}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i^\top(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1}))\boldsymbol{x}_i + \mu(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1})$ which only computes the inner-product $\boldsymbol{x}_i^\top(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1})$ without explicitly computing $\bar{\boldsymbol{H}}$. So the computational cost of each stochastic gradient associated with $\bar{\boldsymbol{H}}$ is actually much cheaper than that of computing stochastic gradient of $\nabla F(\boldsymbol{\theta}_{t-1})$, as the former only involves vector products and the later one is usually complicated, *e.g.* the exponential computation in logistic regression.

**Algorithm 3** Hybrid Stochastic-Deterministic Minibatch Proximal Gradient (HSDMPG) on the generic loss.

```
Input: Regularization constant $\gamma$ and initialization $\boldsymbol{\theta}_0$.
for $t = 1, 2, \ldots, T$ do
    (S1) Construct a finite-sum quadratic function $\boldsymbol{Q}_{t-1}(\boldsymbol{\theta})$ in Eqn. (6) to approximate $F(\boldsymbol{\theta})$ at $\boldsymbol{\theta}_{t-1}$.
    (S2) Run Algorithm 1 with regularization constant $\gamma$ and initialization $\boldsymbol{\theta}_{t-1}$ to minimize the finite-sum function $\boldsymbol{Q}_{t-1}(\boldsymbol{\theta})$ such that $\boldsymbol{Q}_{t-1}(\boldsymbol{\theta}_t) \leq \min_{\boldsymbol{\theta}} \boldsymbol{Q}_{t-1}(\boldsymbol{\theta}) + \varepsilon'_t$.
end for
Output: $\boldsymbol{\theta}_T$.
```

## 4.2 Convergence and Complexity Analysis

Here we also analyze HSDMPG for generic optimization problems under finite-sum and online settings in turn.

### 4.2.1 *Finite-sum setting*

We establish Theorem 3 to guarantee the convergence of Algorithm 3 and analyze its computational complexity. See Appendix C.1 for a proof of this main result.

**Theorem 3.** *Suppose that each loss function $\ell(\boldsymbol{\theta}^\top \boldsymbol{x}, \boldsymbol{y})$ is $L$-smooth and $\sigma$-strongly convex w.r.t. $\boldsymbol{\theta}^\top \boldsymbol{x}$. Consider the following two cases:*
*(1) when the empirical risk $\frac{1}{n}\sum_{i=1}^{n}\ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i)$ is $\mu$-strongly convex, we do not impose any regularization, namely $\tau = 0$;*
*(2) when the empirical risk $\frac{1}{n}\sum_{i=1}^{n}\ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i)$ is only convex, we impose the regularization $\frac{\mu}{2}\|\boldsymbol{\theta}\|_2^2$, namely $\tau = 1$.*

*For both cases, by setting $\varepsilon'_t = \frac{\sigma}{2L}\exp\left(-\frac{\sigma}{2L}t\right)$, with probability at least $1-\delta$ the sequence $\{\boldsymbol{\theta}_t\}$ produced by Algorithm 3 satisfies*

$$\mathbb{E}\left[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^*)\right] \leq \exp\Big(-\frac{\sigma t}{2L}\Big)\big(1 + F(\boldsymbol{\theta}_0) - F(\boldsymbol{\theta}^*)\big),$$

*where the expectation is taken on the randomness of sampling minibatch $\mathcal{S}_t$ to construct the inner subproblem (5) and the randomness of SVRG to solve the subproblem (5) in Algorithm 1. Suppose that the assumptions in Corollary 1 hold. Then with probability at least $1-\delta$, the IFO complexity of Algorithm 3 to achieve $\mathbb{E}\left[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^*)\right] \leq \epsilon$ is of the order*

$$\mathcal{O}\Big(\widetilde{\kappa}(\kappa + \log(d))\log^3\left(\frac{1}{\epsilon}\right) + \frac{\widetilde{\kappa}\nu^2}{\epsilon}\bigwedge \widetilde{\kappa} n \log^2\left(\frac{1}{\epsilon}\right)\Big).$$

*where $\kappa = \frac{L}{\mu}$ and $\widetilde{\kappa} = \frac{L}{\sigma}$.*

Theorem 3 suggests that the objective $F(\boldsymbol{\theta}_t)$ converges linearly to the optimum $F(\boldsymbol{\theta}^*)$ with rate $\exp(-\frac{\sigma}{2L})$. Note that $\sigma$ is the strong convexity parameter of the loss function $\ell(\boldsymbol{\theta}^\top \boldsymbol{x}, \boldsymbol{y})$ w.r.t. $\boldsymbol{\theta}^\top \boldsymbol{x}$ instead of $\boldsymbol{\theta}$ which is usually independent of data scale for widely used loss functions such as the least squared loss and logistic loss [49], and thus leads to fast outer-loop convergence rate.

Now, we compare our IFO complexity with other algorithms. By ignoring the logarithmic factor $\log(d)$ and the small constant $\nu$, our IFO complexity becomes

$$\mathcal{O}\Big(\widetilde{\kappa}\kappa \log^3\left(\frac{1}{\epsilon}\right) + \frac{\widetilde{\kappa}}{\epsilon}\bigwedge \widetilde{\kappa} n \log^2\left(\frac{1}{\epsilon}\right)\Big).$$

Compared with the methods listed in Table 1, one can observe that for generic strongly convex problems, HSDMPG enjoys lower computational complexity than all the compared algorithms except (averaged accelerated) SGD and SCSG for large-scale learning problems where the sample number $n$ is sufficiently large to satisfy the conditions in the third column of Table 1. To compare with SGD and SCSG whose IFO complexity are respectively $\mathcal{O}\big(\frac{1}{\mu\epsilon}\big)$ and $\mathcal{O}\left((n \wedge \frac{\kappa}{\epsilon} + \kappa)\log\left(\frac{1}{\epsilon}\right)\right)$, we need to discuss the condition numbers $\kappa = \frac{L}{\mu}$ and $\widetilde{\kappa} = \frac{L}{\sigma}$. When each individual sample $\boldsymbol{x}_i$ has bounded norm, then $\sigma$ is usually much larger than $\mu$ in the classical and important problems, *e.g.* logistic and softmax regression. Specifically, the strong convexity parameter $\mu$ of the risk function $F$ is typically set of the order $\mathcal{O}\big(1/\sqrt{n}\big)$ so as to match the intrinsic excess error. Taking logistic regression as an example, we have $\sigma = \inf_{\boldsymbol{\theta}\in\Theta, i} \min_i \nabla^2_{\boldsymbol{\theta}^\top \boldsymbol{x}_i} \ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i)) = \inf_{\boldsymbol{\theta}\in\Theta, i} \frac{1}{1+\exp(-\boldsymbol{x}_i^T\boldsymbol{\theta})} \geq \inf_{\boldsymbol{\theta}\in\Theta} \frac{1}{1+\exp(r\|\boldsymbol{\theta}\|_2)}$, where the last inequity uses our assumption $\|\boldsymbol{x}_i\|_2 \leq r$. Since in most problem, the diameter of the convex set $\Theta$ is not related with the data scale $n$, and thus $\frac{1}{1+\exp(r\|\boldsymbol{\theta}\|_2)}$ is often larger than $\mathcal{O}\big(1/\sqrt{n}\big)$ for (moderately) large-scale problems. So we have $\kappa > \widetilde{\kappa}$. This also holds for softmax regression. Under this case, HSDMPG improves over SGD by a factor at least $\mathcal{O}\big(\frac{\kappa}{\widetilde{\kappa}} \wedge \frac{1}{\widetilde{\kappa}\epsilon}\big)$, and improves the factor $\mathcal{O}\left(\frac{\kappa}{\epsilon}\right)$ in SCSG to $\mathcal{O}\left(\frac{\widetilde{\kappa}}{\epsilon}\right)$. When the values of samples obey i.i.d. sub-exponential random variables, then $\sigma = \inf_{\boldsymbol{\theta}\in\Theta, i} \frac{1}{1+\exp(-\boldsymbol{x}_i^T\boldsymbol{\theta})} \approx \frac{1}{1+\exp(C\|\boldsymbol{\theta}\|_2 \log d)}$ can be very small for large $d$, where $C$ is a constant. In this case, $\kappa$ could satisfy $\kappa < \widetilde{\kappa}$, and HSDMPG may not beat SGD and SCSG. It should be mentioned that though this work and averaged SGD [33] need to assume $\|\boldsymbol{x}_i\|_2 \leq r$ $(\forall i)$, this assumption holds for most commonly used data, *e.g.* images, speech signals, and medical data. This means that our method is superior over SGD and SCSG on the real-world data. From Table 1, one can observe that if the problem dimension $d$ obeys $\frac{\kappa\epsilon\xi^3}{\widetilde{\kappa}^2} \leq \mathcal{O}(d)$, then HSDMPG is also more efficient than averaged SGD. These results show the advantages HSDMPG in solving large-scale strongly-convex learning problems. Theorem 3 also shows that HSDMPG improves the complexity $\mathcal{O}\big(\widetilde{\kappa}\kappa\sqrt{s\log(d)}\log^3\left(\frac{1}{\epsilon}\right) + \big(1 + \frac{\kappa^3 \log^{1.5}(d)}{s^{1.5}}\big)\frac{\widetilde{\kappa}\nu^2}{\epsilon} \bigwedge \big(1 + \frac{\kappa \log^{0.5}(d)}{s^{0.5}}\big)\widetilde{\kappa}^3 n \log^2\left(\frac{1}{\epsilon}\right)\big)$ in our previous work [34] which has been discussed in Sec. 3.2.

Finally we consider a realistic case where the optimization error of problem (1) matches the intrinsic excess error bound $\mathcal{O}(1/\sqrt{n})$. For this case, as discussed at the end of Sec. 3.2, the regularization parameter $\mu$ should be set of the order $\mu = \mathcal{O}(1/\sqrt{n})$ to balance the estimation error. As a result, the condition number $\kappa$ could scale as large as $\mathcal{O}(\sqrt{n})$. The following corollary substantializes the IFO complexity bound in Theorem 3 to such a setting. See Appendix C.2 for the proof of this result.

**Corollary 3.** *Suppose that the assumptions in Theorem 3 hold. For both cases (1) and (2) in Theorem 1, with probability at least $1-\delta$, the IFO complexity of* HSDMPG *on the generic loss to achieve $\mathbb{E}[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^*)] \leq \frac{1}{\sqrt{n}}$ is of order $\mathcal{O}\big(n^{0.5}\log^3(n) + \nu^2 n^{0.5}\big)$.*

Corollary 3 shows that for generic convex loss, the IFO complexity of HSDMPG to attain the $\mathcal{O}\big(1/\sqrt{n}\big)$ intrinsic excess error is of the order $\mathcal{O}\left(n^{0.5}\log^3(n)\right)$, where we ignore the constant $\nu^2$ since as aforementioned in Sec. 3.2.1, $\nu^2$ is much smaller than $1/\mu = \mathcal{O}\left(n^{0.5}\right)$. This shows that HSDMPG is able to achieve nearly optimal generalization with less than a single pass over data. Compared with the complexity bound for the quadratic loss, such a more general IFO complexity bound of HSDMPG only comes at the cost of a slightly increased overhead on the logarithmic factor, *i.e.*, from $\log^{1.5}(n)$ for the quadratic case to the $\log^{2.25}(n)$ for generic convex loss. Similar to the observations in the quadratic case, from results in Table 1 one can observe that all the considered state-of-the-art methods need to process the entire data at least one pass to achieve the desired optimization error for generic convex loss. All in all, the established theoretical results for both quadratic and non-quadratic loss functions showcase the benefit of HSDMPG for efficient optimization of large-scale learning problems with near-optimal generalization.

### 4.2.2 Online setting

Now we provide the analysis results of HSDMPG for the generic online problems. Our main results are stated in Theorem 4 with proof in Appendix C.3.

**Theorem 4.** *Assume that each loss $\ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i)$ is L-smooth and $\sigma$-strongly convex w.r.t. $\boldsymbol{\theta}^\top \boldsymbol{x}_i$, and $\sup_{\boldsymbol{\theta}\in\Theta}\mathbb{E}_i[\|\boldsymbol{H}^{-1/2}(\nabla F(\boldsymbol{\theta}) - \nabla \ell_i(\boldsymbol{\theta}))\|_2^2] \leq \nu^2$, where the set $\Theta$ contains the sequence $\{\boldsymbol{\theta}_t\}_{t=0}^T$ produced by Algorithm 1. Consider the following two cases:*
*(1) when the population risk $\mathbb{E}[\ell(\boldsymbol{\theta}^\top \boldsymbol{x}, \boldsymbol{y}; \pi)]$ is $\mu$-strongly convex, we do not impose any regularization, where $\tau = 0$;*
*(2) when the population risk $\mathbb{E}[\ell(\boldsymbol{\theta}^\top \boldsymbol{x}, \boldsymbol{y}; \pi)]$ is only convex, we impose the regularization $\frac{\mu}{2}\|\boldsymbol{\theta}\|_2^2$, where $\tau = 1$.*
*For both cases, with probability at least $1-\delta$ the sequence $\{\boldsymbol{\theta}_t\}$ produced by Algorithm 3 satisfies*

$$\mathbb{E}\left[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^*)\right] \leq \exp\Big(-\frac{\sigma t}{2L}\Big)\big(1 + F(\boldsymbol{\theta}_0) - F(\boldsymbol{\theta}^*)\big),$$

*where the expectation is taken on the randomness of sampling minibatch $\mathcal{S}_t$ to construct the inner subproblem (5) and the randomness of SVRG to solve the subproblem (5) in Algorithm 1. Suppose that the assumptions in Corollary 1 hold. Then by setting $\kappa = \frac{L}{\mu}$ with probability at least $1-\delta$ the IFO complexity of Algorithm 3 to achieve $\mathbb{E}\left[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^*)\right] \leq \epsilon$ is of the order $\mathcal{O}\big(\frac{L\kappa}{\sigma}\log^3\left(\frac{1}{\epsilon}\right) + \frac{L\nu^2}{\sigma\epsilon}\big)$.*

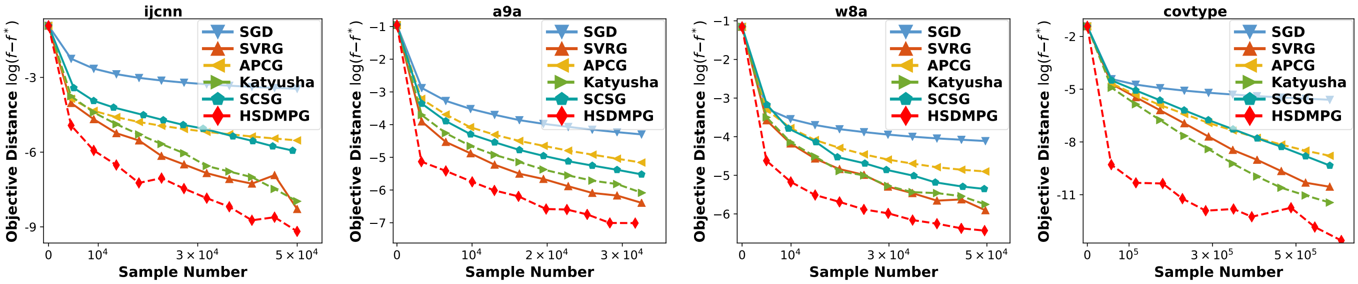


Fig. 2: Single-epoch processing: stochastic gradient algorithms process data a single pass on quadratic problems.

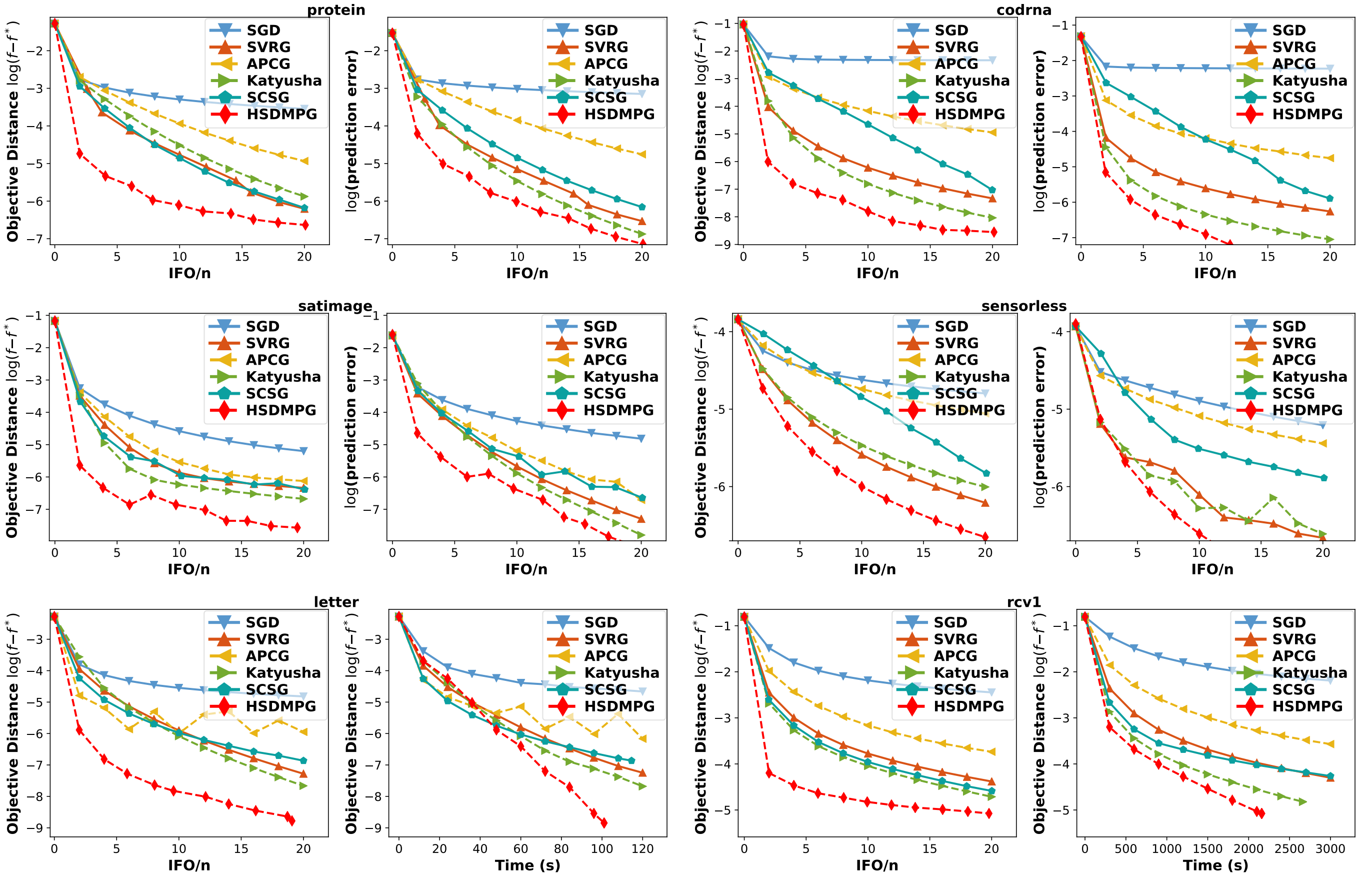


Fig. 3: Multi-epoch processing: stochastic gradient algorithms process data multiple pass on quadratic problems.

Theorem 4 shows that with almost the same assumptions and parameter settings as finite-sum setting, HSDMPG also enjoys linear convergence rate for online-setting which is the same as finite-sum setting. This demonstrates the superior transferability of HSDMPG. For the computational complexity, similar to Sec. 4.2.1, by ignoring the constant $\frac{L}{\sigma}$, the IFO complexity of HSDMPG is

$$\mathcal{O}\Big(\kappa\log^3\Big(\frac{1}{\epsilon}\Big)+\frac{\nu^2}{\epsilon}\Big).$$

In this way, HSDMPG improves the IFO complexity $\mathcal{O}\big(\frac{1}{\mu\epsilon}\big)$ of SGD by a factor of $\mathcal{O}(\kappa\wedge\frac{1}{\epsilon})$, and is also more efficient than SCSG which has IFO complexity $\mathcal{O}\left(\frac{\kappa}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)$ since $\nu^2$ satisfies $\kappa\geq\mathcal{O}(\nu^2)$ theoretically and is empirically shown to be much smaller than $\kappa$ in Figure 1.

## 5 Experiments

In this section, we carry out experiments to compare the numerical performance of HSDMPG with several representative stochastic gradient optimization algorithms, including SGD [15], SVRG [18], APCG [29], Katyusha [21] and SCSG [20].

**Test Problems.** We evaluate all the considered algorithms on two sets of strongly-convex learning tasks. The first set is for ridge regression with a least squared loss

$$\ell(\boldsymbol{\theta}^\top\boldsymbol{x}_i,\boldsymbol{y}_i)=\frac{1}{2}\|\boldsymbol{\theta}^\top\boldsymbol{x}_i-\boldsymbol{y}_i\|_2^2,$$

where $\boldsymbol{y}_i$ is the target output of sample $\boldsymbol{x}_i$. In the second setting we consider two classification models: logistic regression with loss

$$\ell(\boldsymbol{\theta}^\top\boldsymbol{x}_i,\boldsymbol{y}_i)=\log\left(1+\exp(-\boldsymbol{y}_i\boldsymbol{\theta}^\top\boldsymbol{x}_i)\right)$$

and multi-class softmax regression with $k$-classification loss

$$\ell(\boldsymbol{\theta}^\top\boldsymbol{x}_i,\boldsymbol{y}_i)=-\sum_{j=1}^{k}\mathbf{1}\{\boldsymbol{y}_i=j\}\log\left(\frac{\exp(\boldsymbol{\theta}_j^\top\boldsymbol{x}_i)}{\sum_{s=1}^{k}\exp(\boldsymbol{\theta}_s^\top\boldsymbol{x}_i)}\right).$$

**Test Datasets.** We run simulations on twelve datasets, including ijcnn, a9a, w8a, covtype, protein, codrna, satimage, sensorless, letter, rcv1, SUSY and HIGGS. All these datasets are provided on the LibSVM website[1]. Their detailed information is summarized in Table 2. From it we can observe that these datasets are different from each other due to their feature dimension, training samples, and class numbers, *etc*. Thus, these testing datasets can well investigate the performance of the proposed algorithm.

TABLE 2: Descriptions of the twelve testing datasets.

| | #class | #sample | #feature | | #class | #sample | #feature |
|---|---|---|---|---|---|---|---|
| ijcnn1 | 2 | 49,990 | 22 | codrna | 2 | 59,535 | 8 |
| a9a | 2 | 32,561 | 123 | satimage | 6 | 4,435 | 36 |
| w8a | 2 | 49,749 | 300 | sensorless | 11 | 58,509 | 48 |
| covtype | 2 | 581,012 | 54 | rcv1 | 2 | 20,242 | 47,236 |
| protein | 3 | 14,895 | 357 | letter | 26 | 10,500 | 16 |
| SUSY | 2 | 5,000,000 | 18 | HIGGS | 2 | 11,000,000 | 28 |

**Experimental Settings.** For HSDMPG, we set the regularization constant $\eta$ in Bregman divergence (4) as $\eta = 2$ suggested by our theory. Then we set the size $s$ of the initial batch $\mathcal{S}$ around $n^{0.75}$ which is theoretically suggested by our previous work [34]. For the minibatch for inner problems, we set initial minibatch size $|\mathcal{S}_1| = 50$ and then follow our theory to exponentially expand size of $\mathcal{S}_t$ with proper exponential rate. The regularization constant in the subproblem (5) is set to be $\gamma = \log(100d)/s$ as suggested by our theory. The optimization error $\varepsilon_t$ in (5) is controlled by allowing SVRG to run 3 epochs and 10 epochs on the two sets of tasks, respectively. Similarly, we control the optimization error $\varepsilon'_t$ in (7) by running SVRG with 3 epochs. Since there is no ground truth on real data, we run FGD sufficiently long until $\|\nabla F(\tilde{\boldsymbol{\theta}})\|_2 \leq 10^{-10}$ and take $F(\tilde{\boldsymbol{\theta}})$ as an approximate optimal value $F(\boldsymbol{\theta}^*)$ for sub-optimality estimation. In the following subsections, $\log(f - f^*)$ in the label of y-axis representss $\log(F(\boldsymbol{\theta}) - F(\boldsymbol{\theta}^*))$ on the training data; $\log(\text{prediction error})$ in Figures 3 and 6 is defined as $\log\left(\frac{1}{n}\sum_{i=1}^{n} \ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i)\right)$ that measures the validation error over test data $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{n}$. Note, the protein, codrna, satimage, and sensorless datasets in Figures 3, have already split their data into training and test data, and their test sample number are respectively 2,871, 271,617, 2,000, and 10,000. Please see Table 2 for a description of the datasets in use. Since SUSY and HIGGS do not have training-test split, we randomly select 90% of the samples as training data and the rest as test data.

## 5.1 Results on Finite-sum Problems

### 5.1.1 Results for the quadratic loss

**Single-epoch evaluation results.** Here we first evaluate well-conditioned quadratic problems such that moderately accurate solution can be obtained after only one epoch of data pass. Such a one epoch setting usually occurs in online learning. Towards this goal, we set the regularization parameter $\mu = 0.01$ to make the quadratic problems well-conditioned. From Figure 2, one can observe that HSDMPG exhibits much better convergence behavior than the considered baselines, though most algorithms can achieve small optimization error after one epoch processing of data. This confirms the theoretical predictions in Corollaries 1 and 2 that HSDMPG is cheaper in IFO complexity than SGD and variance-reduced algorithms, *e.g.* SVRG and SCSG, when the data scale is large.

**Multi-epoch evaluation results**. For more challenging problems, an algorithm usually requires multiple cycles of data processing to achieve accurate optimization. Here we reset the regularization strength parameter in quadratic problems as $\mu = 10^{-4}$ for generating more challenging optimization tasks. As shown in Figure 3, one can again observe that HSDMPG converges faster than all the compared algorithms in terms of IFO complexity. Particularly, we compare both IFO complexity and wall-clock running time on the letter and rcv11 datasets. The convergence curves under these two metrics consistently show the superior computational efficiency of HSDMPG to the considered state-of-the-arts on large-scale learning tasks, which well support the theoretical predictions in Corollaries 1 and 2.

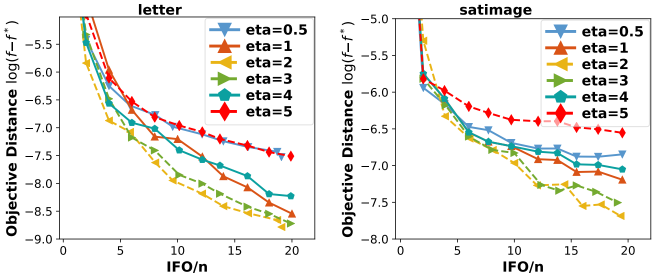


Fig. 4: Investigation of the effects of the regularization constant $\eta$ in the Bregman divergence in Eqn. (4) to the performance of HSDMPG. The test problems are quadratic problems with with regularization constant $\mu = 10^{-4}$ on letter and satimage.

**Robustness evaluation results**. We also investigate the effects of the regularization constant $\eta$ in the Bregman divergence (see Eqn. (4)) to the performance of HSDMPG. Towards this end, we respectively set $\eta$ as $0.5, 1, 2, \cdots, 5$ and run HSDMPG on quadratic problems with regularization constant $\mu = 10^{-4}$. From Figure 4, one can observe that when $\eta = 1, 2$ and 3, HSDMPG is relatively stable, which demonstrates the robustness of HSDMPG. This is because large $\eta$ will hinder the optimization progress, since it encourages the current solution and the previous one to be close; while small $\eta$ allows too aggressive update and could also leads to unsatisfactory performance. Besides, by comparison, one can also observe that HSDMPG with $\eta = 2$ achieves faster convergence speed than HSDMPG with $\eta = 1$. This result is consistent with our Corollary 1 in which we show that HSDMPG with $\eta = 2$ in this work has lower computational complexity than HSDMPG with $\eta = 1$ in our previous work.

### 5.1.2 Results for the non-quadratic loss

Here we investigate the convergence performance of the proposed HSDMPG on non-quadratic convex loss functions. Specifically, we evaluate all the compared algorithms on logistic regression and its multi-classes version, *i.e.* softmax regression, in which their regularization modulus parameters are set as $\mu = 0.01$. Figure 5 reports the running time evolving curves which can accurately reflects the efficiency of an algorithm. These results show that HSDMPG converges significantly faster than the baseline algorithms for the considered non-quadratic loss functions, which well support the predictions in Theorem 3 and Corollary 3 that HSDMPG has lower IFO complexity than the state-of-the-arts in the regimes where data scale is large. This set of results also demonstrates the effectiveness of our sequential quadratic-approximation approach for extending the attractive computational complexity guarantees on quadratic loss to generic convex loss.

## 5.2 Results on Online Problems

Finally, we evaluate our algorithm on the online strongly convex problems. We use two large-scale datasets, namely, SUSY and

[1] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

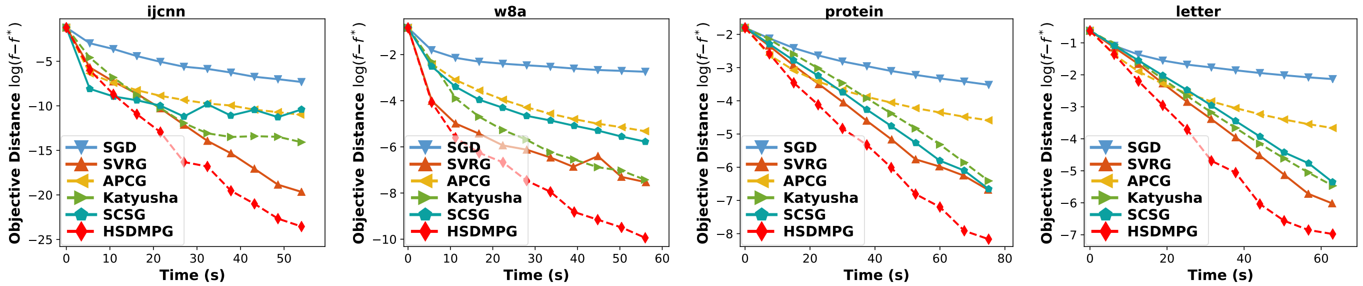


Fig. 5: Multi-epoch processing (about 8 epochs): stochastic gradient algorithms process data multiple pass on logistic regression problems (ijcnn and w08) and softmax regression problems (protein and letter).

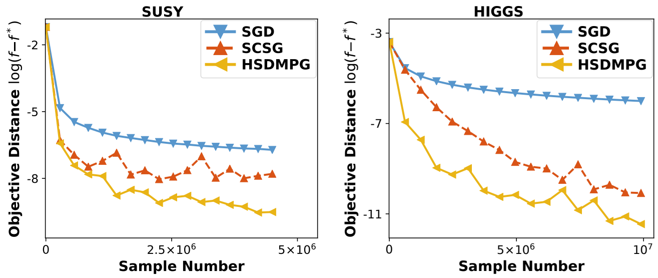


Fig. 6: Evaluation under online setting: stochastic gradient algorithms process data a single pass on quadratic problems (SUSY) and logistic regression problems (HIGGS).

HIGGS in Table 2 in which samples are of the order millions, to simulate online setting. Specifically, as aforementioned, we first randomly permute the samples in the dataset, and then randomly select 90% and 10% for training and test data, respectively. For the algorithm, at each iteration it samples a minibatch of training data under without replacement sampling, and only processes the data with one single pass. Meanwhile, the algorithm evaluates the solution at each iteration on the test data which is randomly selected and can well approximate the population risk. In this way, this process can well mimic the online setting. We compare our algorithm with SGD and SCSG which have online versions, and do not compare other algorithms as they have no online versions. For evaluation, we use SUSY and HIGGS to respectively construct a quadratic and logistic regression problem where their regularization modulus parameters are $\mu = 0.01$.

Figure 6 reports the convergence results of the compared algorithms. From the results, one can observe that HSDMPG converges significantly faster than the baseline algorithms, including SGD and SCSG, on both quadratic and logistic regression problems. These results well support the predictions in our theory for online settings that HSDMPG is of higher efficiency than the state-of-the-arts. Moreover, these results under online setting and the results under finite-sum setting are consistent and demonstrate the advantages and robustness of our HSDMPG.

## 6 Conclusions

We proposed HSDMPG as a hybrid stochastic-deterministic minibach proximal gradient method for strongly convex finite-sum and online problems. Under finite-sum setting, for quadratic loss, we have shown that HSDMPG enjoys provably lower computational complexity than prior state-of-the-art SVRG algorithms in large-scale settings. Particularly, to attain the optimization error $\epsilon = \mathcal{O}\big(1/\sqrt{n}\big)$ at the order of intrinsic excess error bound of ERM which is sufficient for generalization, the stochastic gradient complexity of HSDMPG is dominated by $\mathcal{O}(n^{0.5})$ (up to logarithmic factors). To our best knowledge, HSDMPG for the first time achieves nearly optimal generalization in less than a single pass over data. Almost identical computational complexity guarantees hold for an extension of HSDMPG to generic strongly convex loss functions via sequential quadratic approximation. Besides, we extend HSDMPG from finite-sum setting to online setting and show its higher efficiency than prior state-of-the-arts. Extensive numerical results demonstrate the substantially improved computational efficiency of HSDMPG over the prior methods.

## Acknowledgements

The authors sincerely thank the anonymous reviewers for their constructive comments on this work. Xiao-Tong Yuan would like to acknowledge the partial support from National Key Research and Development Program of China under Grant No. 2018AAA0100400 and Natural Science Foundation of China (NSFC) under Grant No.61876090 and No.61936005. Zhouchen Lin is supported by Key-Area Research and Development Program of Guangdong Province (No. 2019B121204008), NSF China (grant no.s 61625301 and 61731018), Major Scientific Research Project of Zhejiang Lab (grant no.s 2019KB0AC01 and 2019KB0AB02), and Beijing Academy of Artificial Intelligence.

## References

[1] V. Monga, *Handbook of Convex Optimization Methods in Imaging Science*, vol. 1, Springer, 2017.

[2] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2008.

[3] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[4] P. Zhou, C. Lu, J. Feng, Z. Lin, and S. Yan, "Tensor low-rank representation for data recovery and clustering," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2019.

[5] P. Zhou and J. Feng, "Outlier-robust tensor pca," 2017.

[6] D. Palomar and Y. Eldar, *Convex optimization in signal processing and communications*, Cambridge university press, 2010.

[7] J. Mattingley and S. Boyd, "Real-time convex optimization in signal processing," *IEEE Signal processing magazine*, vol. 27, no. 3, pp. 50–61, 2010.

[8] C. Chi, W. Li, and C. Lin, *Convex optimization for signal processing and communications: from fundamentals to applications*, CRC press, 2017.

[9] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[10] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, vol. 87, Springer Science & Business Media, 2013.

[11] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical learning with sparsity: the lasso and generalizations*, CRC press, 2015.

[12] V. Boyarshinov, *Machine learning in computational finance*, Rensselaer Polytechnic Institute, 2005.

[13] T. Pennanen, "Introduction to convex optimization in financial markets," *Mathematical programming*, vol. 134, no. 1, pp. 157–186, 2012.

[14] M. A. Cauchy, "Méthode générale pour la résolution des systèmes d'équations simultanées," *Comptesrendus des séances de l'Académie des sciences de Paris*, vol. 25, pp. 536–538, 1847.

[15] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[16] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.

[17] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Proc. Conf. Neural Information Processing Systems*, 2014, pp. 1646–1654.

[18] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. Conf. Neural Information Processing Systems*, 2013, pp. 315–323.

[19] H. Lin, J. Mairal, and Z. Harchaoui, "A universal catalyst for first-order optimization," in *Proc. Conf. Neural Information Processing Systems*, 2015, pp. 3384–3392.

[20] L. Lei and M. Jordan, "Less than a single pass: Stochastically controlled stochastic gradient," in *Artificial Intelligence and Statistics*, 2017, pp. 148–156.

[21] Z. Allen-Zhu, "Katyusha: The First Direct Acceleration of Stochastic Gradient Methods," in *ACM SIGACT Symposium on Theory of Computing*, 2017.

[22] G. Lan, Z. Li, and Y. Zhou, "A unified variance-reduced accelerated gradient method for convex optimization," in *Proc. Conf. Neural Information Processing Systems*, 2019, pp. 10462–10472.

[23] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Proc. Conf. Neural Information Processing Systems*, 2008, pp. 161–168.

[24] V. Vapnik, *Estimation of dependences based on empirical data*, Springer Science & Business Media, 2006.

[25] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan, "Stochastic convex optimization.," in *Conf. on Learning Theory*, 2009.

[26] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*, Cambridge university press, 2014.

[27] O. Shamir, "Making gradient descent optimal for strongly convex stochastic optimization," *CoRR abs/1109.5647*, 2011.

[28] S. Shalev-Shwartz and T. Zhang, "Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization," in *Proc. Int'l Conf. Machine Learning*, 2014, pp. 64–72.

[29] Q. Lin, Z. Lu, and L. Xiao, "An accelerated proximal coordinate gradient method," in *Proc. Conf. Neural Information Processing Systems*, 2014, pp. 3059–3067.

[30] Y. Zhang and L. Xiao, "Stochastic primal-dual coordinate method for regularized empirical risk minimization," in *Proc. Int'l Conf. Machine Learning*, 2015, pp. 353–361.

[31] Atsushi A. Nitanda, "Accelerated stochastic gradient descent for minimizing finite sums," in *Artificial Intelligence and Statistics*, 2016, pp. 195–203.

[32] A. Dieuleveut, N. Flammarion, and F. Bach, "Harder, better, faster, stronger convergence rates for least-squares regression," *J. of Machine Learning Research*, vol. 18, no. 1, pp. 3520–3570, 2017.

[33] F. Bach and E. Moulines, "Non-strongly-convex smooth stochastic approximation with convergence rate o (1/n)," in *Proc. Conf. Neural Information Processing Systems*, 2013, pp. 773–781.

[34] P. Zhou and X. Yuan, "Hybrid stochastic-deterministic minibatch proximal gradient: Less-than-single-pass optimization with nearly optimal generalization," in *Proc. Int'l Conf. Machine Learning*, 2020.

[35] L. Bottou, "Stochastic gradient learning in neural networks," *Proceedings of Neuro-Nımes*, vol. 91, no. 8, pp. 12, 1991.

[36] H. Hendrikx, F. Bach, and L. Massoulié, "Asynchronous accelerated proximal stochastic gradient for strongly convex distributed finite sums," *arXiv preprint arXiv:1901.09865*, 2019.

[37] H. Mohammadi, M. Razaviyayn, and M. Jovanović, "Robustness of accelerated first-order algorithms for strongly convex optimization problems," *arXiv preprint arXiv:1905.11011*, 2019.

[38] M. P. Friedlander and M. Schmidt, "Hybrid deterministic-stochastic methods for data fitting," *SIAM Journal on Scientific Computing*, vol. 34, no. 3, pp. A1380–A1405, 2012.

[39] P. Zhou, X. Yuan, and J. Feng, "Efficient stochastic gradient hard thresholding," in *Proc. Conf. Neural Information Processing Systems*, 2018.

[40] P. Zhou, X. Yuan, and J. Feng, "New insight into hybrid stochastic gradient descent: Beyond with-replacement sampling and convexity," in *Proc. Conf. Neural Information Processing Systems*, 2018, pp. 1234–1243.

[41] A. Mokhtari, H. Daneshmand, A. Lucchi, T. Hofmann, and A. Ribeiro, "Adaptive newton method for empirical risk minimization to statistical accuracy," in *Proc. Conf. Neural Information Processing Systems*, 2016, pp. 4062–4070.

[42] A. Mokhtari and A. Ribeiro, "First-order adaptive sample size methods to reduce complexity of empirical risk minimization," in *Proc. Conf. Neural Information Processing Systems*, 2017, pp. 2060–2068.

[43] O. Shamir, N. Srebro, and T. Zhang, "Communication-efficient distributed optimization using an approximate newton-type method," in *Proc. Int'l Conf. Machine Learning*, 2014, pp. 1000–1008.

[44] M. Hardt, B. Recht, and Y. Singer, "Train faster, generalize better: Stability of stochastic gradient descent," in *Proc. Int'l Conf. Machine Learning*, 2016.

[45] P. Zhou and J. Feng, "Understanding generalization and optimization performance of deep cnns," in *Proc. Int'l Conf. Machine Learning*, 2018.

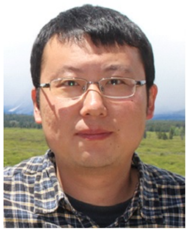
[46] P. Zhou and J. Feng, "Empirical risk landscape analysis for understanding deep neural networks," in *Int'l Conf. Learning Representations*, 2018.

[47] O. Bousquet and A. Elisseeff, "Stability and generalization," *J. of Machine Learning Research*, vol. 2, no. Mar, pp. 499–526, 2002.

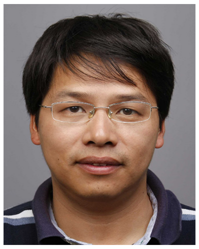
[48] V. Feldman and J. Vondrak, "High probability generalization bounds for uniformly stable algorithms with nearly optimal rate," in *Conf. on Learning Theory*, 2019, pp. 1270–1279.

[49] X. Yuan and P. Li, "On convergence of distributed approximate newton methods: Globalization, sharper bounds and beyond," *J. of Machine Learning Research*, 2020.

**Pan Zhou** received Master Degree in computer science from Peking University in 2016 and obtained Ph.D. Degree in computer science from National University of Singapore in 2019. Now he is a senior research scientist in Sea AI Lab of Sea group, Singapore. Before, he was also a research scientist in Salesforce. His research interests include computer vision, machine learning, and optimization. He was the winner of the Microsoft Research Asia Fellowship 2018.



**Xiao-Tong Yuan** received the PhD degree in pattern recognition from Chinese Academy of Sciences, in 2009. After graduation, he held various appointments as postdoctoral research associate working in National University of Singapore, Rutgers University, and Cornell University. In 2013, he joined Nanjing University of Information Science & Technology where, he is currently a professor of computer science. His main research interests include machine learning, data mining, and computer vision. He is a member of the IEEE.



**Zhouchen Lin** is currently a professor in School of Electronics Engineering and Computer Science, Peking University. His research interests include computer vision, image processing, machine learning, pattern recognition, and numerical optimization. He is an area chair of CVPR 2014/2016/2019, ICCV 2015, NIPS 2015/2018 and AAAI 2019, and a senior program committee member of AAAI 2016/2017/2018 and IJCAI 2016/2018. He is an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence and the International Journal of Computer Vision. He is an IAPR Fellow and IEEE Fellow.

**Steven C. H. Hoi** is currently the Managing Director of Salesforce Research Asia, and an Associate Professor of Singapore Management University, Singapore. His research interests are machine learning and data mining and their applications to multimedia information retrieval, social media and web mining, and computational finance, etc. He is an IEEE Fellow and ACM Distinguished Member.