Variance Reduction via Primal-Dual Accelerated Dual Averaging for Nonsmooth Convex Finite-Sums

Chaobing Song [1] **Stephen J. Wright** [1] **Jelena Diakonikolas** [1]

Abstract

Structured nonsmooth convex finite-sum optimization appears in many machine learning applications, including support vector machines and least absolute deviation. For the primal-dual formulation of this problem, we propose a novel algorithm called *Variance Reduction via Primal-Dual Accelerated Dual Averaging (*VRPDA^2*)*. In the nonsmooth and general convex setting, VRPDA^2 has the overall complexity $O(nd\log\min\{1/\epsilon,n\}+d/\epsilon)$ in terms of the primal-dual gap, where n denotes the number of samples, d the dimension of the primal variables, and ϵ the desired accuracy. In the nonsmooth and strongly convex setting, the overall complexity of VRPDA^2 becomes $O(nd\log\min\{1/\epsilon,n\}+d/\sqrt{\epsilon})$ in terms of both the primal-dual gap and the distance between iterate and optimal solution. Both these results for VRPDA^2 improve significantly on state-of-the-art complexity estimates—which are $O(nd\log\min\{1/\epsilon,n\}+\sqrt{n}d/\epsilon)$ for the nonsmooth and general convex setting and $O(nd\log\min\{1/\epsilon,n\}+\sqrt{n}d/\sqrt{\epsilon})$ for the nonsmooth and strongly convex setting—with a simpler and more straightforward algorithm and analysis. Moreover, both complexities are better than *lower* bounds for general convex finite-sum optimization, because our approach makes use of additional, commonly occurring structure. Numerical experiments reveal competitive performance of VRPDA^2 compared to state-of-the-art approaches.

1. Introduction

We consider large-scale regularized nonsmooth convex empirical risk minimization (ERM) of linear predictors in machine learning. Let $\boldsymbol{b}_i \in \mathbb{R}^d$, $i=1,2,\ldots,n$, be sample vectors with n typically large; $g_i:\mathbb{R}\to\mathbb{R}$, $i=1,2,\ldots,n$, be possibly *nonsmooth* convex loss functions associated with the linear predictor $\langle \boldsymbol{b}_i,\boldsymbol{x}\rangle$; and $\ell:\mathbb{R}^d\to\mathbb{R}$ be an extended-real-valued, σ-strongly convex ($\sigma\geq 0$) and possibly nonsmooth regularizer that admits an efficiently computable proximal operator. The problem we study is

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) := g(\boldsymbol{x})+\ell(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} g_i(\boldsymbol{b}_i^T\boldsymbol{x})+\ell(\boldsymbol{x}), \quad \text{(P)}$$

where $g(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^n g_i(\boldsymbol{b}_i^T\boldsymbol{x})$. Instances of the nonsmooth ERM problem (P) include ℓ_1-norm and ℓ_2-norm regularized support vector machines (SVM) and least absolute deviation. For practicality of our approach, we require in addition that the convex conjugates of the functions g_i, defined by $g_i^*(y_i) := \sup_{z_i}(z_iy_i - g_i(z_i))$, admit efficiently computable proximal operators. (The examples mentioned above have this property.) From the statistical perspective, nonsmoothness in the loss function is essential for obtaining a model that is both tractable and robust. But from the optimization viewpoint, nonsmooth optimization problems are intrinsically more difficult to solve. On one hand, the lack of smoothness in g precludes the use of black-box first-order information to obtain efficient methods. On the other hand, the use of structured composite optimization methods that rely on the proximal operator of g is out of question here too, because the proximal operator of the sum $\frac{1}{n}\sum_{i=1}^n g_i({\boldsymbol{b}_i}^T\boldsymbol{x})$ may not be efficiently computable w.r.t. $\boldsymbol{x}$, even when the proximal operators of the individual functions $g_i(\cdot)$ are.

Driven by applications in machine learning, computational statistics, signal processing, and operations research, the nonsmooth problem (P) and its variants have been studied for more than two decades. There have been two main lines of work: deterministic algorithms that exploit the underlying simple primal-dual structure to improve efficiency (i.e., dependence on the accuracy parameter ϵ) and randomized algorithms that exploit the finite-sum structure to improve scalability (i.e., dependence on the number of samples n).

Exploiting the primal-dual structure. A naïve approach for solving (P) would be subgradient descent, which requires access to subgradients of $g(\boldsymbol{x})$ and $\ell(\boldsymbol{x})$. To find a solution $\boldsymbol{x}$ with $f(\boldsymbol{x})-f(\boldsymbol{x}^*)\leq\epsilon$, where $\boldsymbol{x}^*$ is an optimal solution

[1]Department of Computer Sciences, University of Wisconsin-Madison, Madison, WI. Correspondence to: Chaobing Song <chaobing.song@wisc.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

of (P) and $\epsilon > 0$ is the desired accuracy, the subgradient method requires $O(1/\epsilon^2)$ iterations for the nonsmooth convex setting. This complexity is high, but it is also the best possible if we are only allowed to access "black-box" information of function value and subgradient. To obtain improved complexity bounds, we must consider approaches that exploit structure in (P). To begin, we note that (P) admits an explicit and simple primal-dual reformulation:

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathbb{R}^d}\big\{f(\boldsymbol{x}) = \max_{\boldsymbol{y}\in\mathbb{R}^n} L(\boldsymbol{x},\boldsymbol{y})\big\},\\ &L(\boldsymbol{x},\boldsymbol{y}) := \langle \boldsymbol{B}\boldsymbol{x},\boldsymbol{y}\rangle - g^*(\boldsymbol{y}) + \ell(\boldsymbol{x}), \end{aligned} \tag{PD}$$

where $\boldsymbol{B} = \frac{1}{n}[\boldsymbol{b}_1,\boldsymbol{b}_2,\dots,\boldsymbol{b}_n]^T$, $g^*(\boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^n g_i^*(y_i)$ with the convex conjugate functions $g_i^*(\cdot)$ satisfying $g_i(\boldsymbol{b}_i^T\boldsymbol{x}) = \sup_{y_i}\{y_i\langle \boldsymbol{b}_i,\boldsymbol{x}\rangle - g_i^*(y_i)\}$. The nonsmooth loss $\boldsymbol{g}(\boldsymbol{x})$ in (P) is thereby decoupled into a bilinear term $\langle \boldsymbol{B}\boldsymbol{x},\boldsymbol{y}\rangle$ and a separable function $g^*(\boldsymbol{y})$ that admits an efficiently computable proximal operator. Due to the possible nonsmoothness of $g(\boldsymbol{x})$, we can assume only that $L(\boldsymbol{x},\boldsymbol{y})$ is concave *w.r.t.* $\boldsymbol{y}$—but not strongly concave. Therefore, Problem (PD) is σ-strongly convex-(general) concave ($\sigma \geq 0$).

By adding a strongly convex regularizer to the dual variable of (PD), Nesterov (2005b) optimized a smoothed variant of (P) using acceleration, thus improving the complexity bound from $O(1/\epsilon^2)$ to $O(1/\epsilon)$. Later, Nemirovski and Nesterov, respectively, showed that extragradient methods such as mirror-prox (Nemirovski, 2004) and dual extrapolation (Nesterov, 2007) can obtain the same $O(1/\epsilon)$ complexity bound for (PD) directly, without the use of smoothing or Nesterov's acceleration. (Extragradient methods perform updates twice per iteration, for both primal and dual variables.) Chambolle & Pock (2011) introduced an (extrapolated) primal-dual hybrid gradient (PDHG) method to obtain the same $O(1/\epsilon)$ complexity, using an extrapolation step on either the primal or dual variable rather than an extragradient step. Thus, PDHG needs to update primal and dual variables just once per iteration. All three kinds of methods have been extensively studied from different perspectives (Nesterov, 2005a; Chen et al., 2017; Tran-Dinh et al., 2018; Song et al., 2020b; Diakonikolas et al., 2020). For large n, the focus has been on randomized variants with low per-iteration cost (Zhang & Lin, 2015; Alacaoglu et al., 2017; Tan et al., 2018; Chambolle et al., 2018; Carmon et al., 2019; Lei et al., 2019; Devraj & Chen, 2019; Alacaoglu et al., 2020).

Exploiting the finite-sum structure. The deterministic methods discussed above have per-iteration cost $O(nd)$, which can be prohibitively high for large n. There has been much work on randomized methods whose per-iteration cost is independent of n. To be efficient, the iteration count of such methods cannot increase too much over the deterministic methods. A major development in the past decade of research has been the use of *variance reduction* in randomized optimization algorithms, which reduces the per-iteration cost and improves the overall complexity. For the variant of Problem (P) in which $g(\boldsymbol{x})$ is *smooth*, there exists a vast literature on developing efficient finite-sum solvers; see for example Roux et al. (2012); Johnson & Zhang (2013); Lin et al. (2014); Zhang & Lin (2015); Allen-Zhu (2017); Zhou et al. (2018); Lan et al. (2019); Song et al. (2020a). The *Variance Reduction via Accelerated Dual Averaging (*VRADA*)* algorithm of Song et al. (2020a) matches all three lower bounds from Woodworth & Srebro (2016); Hannah et al. (2018) for the smooth and (general/ill-conditioned strongly/well-conditioned strongly) convex settings, using a simple, unified algorithm description and convergence analysis. As discussed in Song et al. (2020a), the efficiency, simplicity, and unification of VRADA are due to a novel initialization strategy and to randomizing *accelerated dual averaging* rather than *accelerated mirror descent* (as was done in Allen-Zhu (2017)). These results provide the main motivation for our current work.

When the loss function is nonsmooth, classical variance reduction approaches such as SVRG and SAGA (Johnson & Zhang, 2013; Defazio et al., 2014) are no longer applicable. Allen-Zhu & Hazan (2016); Allen-Zhu (2017) propose to smoothen and regularize (P) and then apply existing finite-sum solvers, such as Katyusha. As shown by Allen-Zhu (2017), in the nonsmooth and general convex setting, the resulting overall complexity is improved from $O\big(\frac{nd}{\epsilon}\big)$ to $O\big(nd\log\min\{\frac{1}{\epsilon},n\} + \frac{\sqrt{n}d}{\epsilon}\big)$; in the nonsmooth and strongly convex setting, it is improved from $O\big(\frac{nd}{\sqrt{\epsilon}}\big)$ to $O\big(nd\log\min\{\frac{1}{\epsilon},n\} + \frac{\sqrt{n}d}{\sqrt{\epsilon}}\big)$. Both of these improved complexity results match the lower bounds of Woodworth & Srebro (2016) for *general* nonsmooth finite-sums when ϵ is small. However, the smoothing and regularization require tuning of additional parameters, which complicates the algorithm implementation. Meanwhile, it is not clear whether the complexity can be further improved to take advantage of the additional ERM structure of (P).

For the nonsmooth ERM problem (P) considered here, and its primal-dual formulation, the literature is much scarcer (Dang & Lan, 2014; Alacaoglu et al., 2017; Chambolle et al., 2018; Carmon et al., 2019; Latafat et al., 2019; Fercoq & Bianchi, 2019; Alacaoglu et al., 2020). All existing methods target (PD) directly and focus on extending the aforementioned deterministic algorithms to this case. Because sampling one element of the finite sum from (P) is reduced to sampling one dual coordinate in (PD), all these methods can be viewed as coordinate variants of the deterministic counterparts. For convenience, we explicitly rewrite (PD) in the following finite-sum primal-dual form:

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathbb{R}^d}\max_{\boldsymbol{y}\in\mathbb{R}^n} L(\boldsymbol{x},\boldsymbol{y})\\ &L(\boldsymbol{x},\boldsymbol{y}) := \frac{1}{n}\sum_{i=1}^n \big(y_i\langle \boldsymbol{b}_i,\boldsymbol{x}\rangle - g_i^*(y_i)\big) + \ell(\boldsymbol{x}). \end{aligned} \tag{FS-PD}$$

Table 1. Overall complexity and per-iteration cost for solving (FS-PD) in the σ-strongly convex-general concave setting ($\sigma \geq 0$). ("—" indicates that the corresponding result does not exist or is unknown.)

Algorithm	General Convex (Primal-Dual Gap)	Strongly Convex (Primal-Dual Gap)	Strongly Convex (Distance to Solution)	Per-Iteration Cost
RPD Dang & Lan (2014)	$O\big(\frac{n^{3/2}d}{\epsilon}\big)$	$O\big(\frac{n^{3/2}d}{\sqrt{\epsilon}}\big)$	—	$O(d)$
SMART-CD Alacaoglu et al. (2017)	$O\big(\frac{nd}{\epsilon}\big)$	—	—	$O(d)$
Carmon et al. (2019)	$O\big(nd + \frac{\sqrt{nd(n+d)}\log(nd)}{\epsilon}\big)$	—	—	$O(n+d)$
SPDHG Chambolle et al. (2018)	$O\big(\frac{nd}{\epsilon}\big)$	—	$O\big(\frac{nd}{\sigma\sqrt{\epsilon}}\big)$ [1]	$O(d)$
PURE-CD Alacaoglu et al. (2020)	$O\big(\frac{n^2d}{\epsilon}\big)$	—	—	$O(d)$
VRPDA^2 (**This Paper**)	$O(nd\log\min\{\frac{1}{\epsilon}, n\} + \frac{d}{\epsilon})$	$O(nd\log\min\{\frac{1}{\epsilon}, n\} + \frac{d}{\sqrt{\sigma\epsilon}})$	$O(nd\log\min\{\frac{1}{\epsilon}, n\} + \frac{d}{\sigma\sqrt{\epsilon}})$	$O(d)$

[1] It is only applicable when ϵ is small enough (see Chambolle et al. (2018, Theorem 5.1)).

Existing approaches. Table 1 compares VRPDA^2 to existing randomized algorithms for solving (FS-PD) in terms of the overall complexity and per-iteration cost under the setting of uniform sampling and general (*i.e.,* not necessarily sparse) data matrix. The algorithms RPD, SMART-CD, SPDHG, and PURE-CD all attain $O(d)$ per-iteration cost, but have overall complexity no better than that of the deterministic algorithms in both the nonsmooth and general/strongly convex settings. Meanwhile, the algorithms of Carmon et al. (2019) perform full-coordinate updates with $O(n+d)$ per-iteration cost and improve the dependence on the dimension d when $n \geq d$. However, the overall dependence on the dominant term n is still not improved, which raises the question of whether it is even possible to simultaneously achieve the low $O(d)$ per-iteration cost and reduce the overall complexity compared to the deterministic algorithms. Addressing this question is the main contribution of our work.

Our contributions. We propose the VRPDA^2 algorithm for (FS-PD) in the σ-strongly convex-general concave setting ($\sigma \geq 0$), which corresponds to the nonsmooth and σ-strongly convex setting of (P) ($\sigma \geq 0$). For both $\sigma = 0$ and $\sigma > 0$, VRPDA^2 has $O(d)$ per-iteration cost and significantly improves the best-known overall complexity results in a unified and simplified way. As shown in Table 1, to find an ϵ-accurate solution in terms of the primal-dual gap, the overall complexity of VRPDA^2 is

$$\begin{cases} O\big(nd\log\big(\min\{\frac{1}{\epsilon}, n\}\big) + \frac{d}{\epsilon}\big), & \text{if } \sigma = 0, \\ O\big(nd\log\big(\min\{\frac{1}{\epsilon}, n\}\big) + \frac{d}{\sqrt{\sigma\epsilon}}\big), & \text{if } \sigma > 0, \end{cases}$$

which is significantly better than any of the existing results for (FS-PD). In particular, we only need $O(nd\log n)$ overall cost to attain an ϵ-accurate solution with $\epsilon = \Omega(\frac{1}{n\log(n)})$. Meanwhile, when ϵ is sufficiently small compared to $1/n$, so that the second term in the bound becomes dominant, the overall complexity ($O(\frac{d}{\epsilon})$ for $\sigma = 0$ and $O(\frac{d}{\sqrt{\sigma\epsilon}})$ for $\sigma > 0$) is independent of n, thus showing a $\Theta(n)$ improvement compared to the deterministic algorithms. To the best of our knowledge, even for smooth g_i's, the improvement of existing algorithms is at most $\Theta(\sqrt{n})$ and is attained by accelerated variance reduction methods such as Katyusha (Allen-Zhu, 2017) and VRADA (Song et al., 2020a).

Comparison to lower bounds. Our results may seem to contradict the iteration complexity lower bounds for composite objectives, which are $\Omega(n + \frac{\sqrt{n}}{\epsilon})$ for nonsmooth and general convex objectives and $\Omega(n + \sqrt{\frac{n}{\sigma\epsilon}})$ for nonsmooth and σ-strongly convex objectives (Woodworth & Srebro, 2016). In Woodworth & Srebro (2016, Section 5.1), the hard instance for proving the lower bounds has the form $f(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x})$—but each f_i is a sum of $k+1$ "simple" terms, each having the form of our g_i's. The complexity in Woodworth & Srebro (2016) for this hard instance is enabled by hiding the individual vectors corresponding to each simple term, an approach that is typical for oracle lower bounds. In their example, $k = \Theta(\frac{1}{\sqrt{n\epsilon}})$, so the total number of simple terms is $nk = \Theta(\frac{\sqrt{n}}{\epsilon})$, which leads to the second term in the lower bound. (The first $\Omega(n)$ term in this lower bound comes from setting $\epsilon = O(\frac{1}{\sqrt{n}})$.) Applying our upper bound for iteration complexity to this hard case, we replace n by $nk = \Theta(\frac{\sqrt{n}}{\epsilon})$ to obtain $O(\frac{\sqrt{n}}{\epsilon}\log(\frac{\sqrt{n}}{\epsilon}))$—higher than the Woodworth & Srebro (2016) lower bound would be if we were to replace n by nk. Thus, our results do not contradict these well known lower bounds.

Remarkably, our upper bounds show that use of the finite-sum primal-dual formulation (FS-PD) can lead not only to improvements in efficiency (dependence on ϵ), as in Nesterov (2005b), but also scalability (dependence on n). As the ERM problem (P) is one of the main motivations for convex finite-sum solvers, it would be interesting to characterize the complexity of the problem class (P) from the aspect of oracle lower bounds and determine whether VRPDA^2 attains optimal oracle complexity. (We conjecture that it does, at

least for small values of ϵ.) Since the primary focus of the current paper is on algorithms, we leave the study of lower bounds for future research.

Our techniques. Our VRPDA^2 algorithm is founded on a new deterministic algorithm *Primal-Dual Accelerated Dual Averaging (*PDA^2*)* for (PD). Similar to PDHG (Chambolle & Pock, 2011), PDA^2 is a primal-dual method with extrapolation on the primal or dual variable. However, unlike PDHG, which is based on mirror-descent-type updates (*a.k.a.* agile updates (Allen-Zhu & Orecchia, 2017)), PDA^2 performs updates of dual averaging-style (Nesterov, 2015) (*a.k.a.* lazy mirror-descent updates (Hazan et al., 2016)).

Our analysis is based on the classical estimate sequence technique, but with a novel design of the estimate sequences that requires careful coupling of primal and dual portions of the gap; see Section 3 for a further discussion. The resulting argument allows us to use a unified parameter setting and convergence analysis for PDA^2 in all the (general/strongly) convex-(general/strongly) concave settings. Thus, by building on PDA^2 rather than PDHG, the design and analysis of VRPDA^2 is unified over the different settings and also significantly simplified. Moreover, the dual averaging framework allows us to use a novel initialization strategy inspired by the VRADA algorithm (Song et al., 2020a), which is key to cancelling the randomized error of order n in the main loop and obtaining our improved results from Table 1. It is worth noting that although PDA^2 can be used in all the (general/strongly) convex-(general/strongly) concave settings, VRPDA^2 is applicable only to the specific (general/strongly) convex-general concave settings that correspond to the nonsmooth and (general/strongly) convex settings of (P). Study of VRPDA^2 in the (general/strongly) convex-strongly concave settings is deferred to future research.

2. Notation and Preliminaries

Throughout the paper, we use $\|\cdot\|$ to denote the Euclidean norm. In the case of matrices $\boldsymbol{B}$, $\|\boldsymbol{B}\|$ is the standard operator norm defined by $\|\boldsymbol{B}\| := \max_{\boldsymbol{x}\in\mathbb{R}^d,\, \|\boldsymbol{x}\|\le 1} \|\boldsymbol{B}\boldsymbol{x}\|$.

In the following, we provide standard definitions and properties that will be used in our analysis. We start by stating the definition of strongly convex functions that captures both strong and general convexity, allowing us to treat both cases in a unified manner for significant portions of the analysis. We use $\bar{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$ to denote the extended real line.

Definition 1. *Given $\sigma \ge 0$, we say that a function $f : \mathbb{R}^d \to \bar{\mathbb{R}}$ is σ-strongly convex, if $\forall \boldsymbol{x}, \hat{\boldsymbol{x}} \in \mathbb{R}^d$, and all $\alpha \in (0,1)$*

$$f((1-\alpha)\boldsymbol{x} + \alpha\hat{\boldsymbol{x}}) \le (1-\alpha)f(\boldsymbol{x}) + \alpha f(\hat{\boldsymbol{x}}) - \frac{\sigma}{2}\alpha(1-\alpha)\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2.$$

When $\sigma = 0$, we say that f is (general) convex.

When f is subdifferentiable at $\boldsymbol{x}$ and $\boldsymbol{g}_f(\boldsymbol{x}) \in \partial f(\boldsymbol{x})$ is any subgradient of f at $\boldsymbol{x}$, where $\partial f(\boldsymbol{x})$ denotes the subdifferential set (the set of all subgradients) of f at $\boldsymbol{x}$, then strong convexity implies that for all $\hat{\boldsymbol{x}} \in \mathbb{R}^d$, we have

$$f(\hat{\boldsymbol{x}}) \ge f(\boldsymbol{x}) + \langle \boldsymbol{g}_f(\boldsymbol{x}), \hat{\boldsymbol{x}} - \boldsymbol{x}\rangle + \frac{\sigma}{2}\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2.$$

Since we work with general nonsmooth convex functions f, we require that their *proximal operators*, defined as solutions to problems of the form $\min_{\boldsymbol{x}\in\mathbb{R}^d} \left\{ f(\boldsymbol{x}) + \frac{1}{2\tau}\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 \right\}$ are efficiently solvable for any $\tau > 0$ and any $\hat{\boldsymbol{x}} \in \mathbb{R}^d$.

Problem definition. As discussed in the introduction, our focus is on Problem (PD) under the following assumption.

Assumption 1. *$g^*(\boldsymbol{y})$ is proper, l.s.c., and γ-strongly convex ($\gamma \ge 0$); $\ell(\boldsymbol{x})$ is proper, l.s.c., and σ-strongly convex ($\sigma \ge 0$); the proximal operators of g^* and ℓ can be computed efficiently; and $\|\boldsymbol{B}\| = R$ for some $R \in (0, \infty)$.*

Observe that since g^* and ℓ are assumed only to be proper, l.s.c., and (strongly) convex, they may contain indicators of closed convex sets in their description. Thus, certain constrained optimization problems are included in the problem class described by Assumption 1. We use $\mathcal{X}$ and $\mathcal{Y}$ to denote the domains of ℓ and g^*, respectively, defined by $\mathcal{X} = \mathrm{dom}(\ell) = \{\boldsymbol{x} : \ell(\boldsymbol{x}) < \infty\}$, $\mathcal{Y} = \mathrm{dom}(g^*) = \{\boldsymbol{y} : g^*(\boldsymbol{y}) < \infty\}$. When $\mathcal{X}, \mathcal{Y}$ are bounded, we use $D_{\mathcal{X}}, D_{\mathcal{Y}}$ to denote their diameters: $D_{\mathcal{X}} = \max_{\boldsymbol{x},\boldsymbol{u}\in\mathcal{X}} \|\boldsymbol{x} - \boldsymbol{u}\|$, $D_{\mathcal{Y}} = \max_{\boldsymbol{y},\boldsymbol{v}\in\mathcal{Y}} \|\boldsymbol{y} - \boldsymbol{v}\|$.

Note that Assumption 1 does not enforce a finite-sum structure of g^* (and g). Thus, for the results that utilize variance reduction, we will make a further assumption.

Assumption 2. *$g^*(\boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^n g_i^*(y_i)$, where each $g_i^*(y_i)$ is convex and has an efficiently computable proximal operator. Further, $\|\boldsymbol{b}_i\| \le R'$, for all $i \in \{1, \ldots, n\}$.*

Recall that $\boldsymbol{B} = \frac{1}{n}[\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n]^T$. Observe that $R = \|\boldsymbol{B}\| \le \frac{1}{n}\left(\sum_{i=1}^n \|\boldsymbol{b}_i\|^2\right)^{1/2} \le \frac{1}{n}\sum_{i=1}^n \|\boldsymbol{b}_i\| \le R'$.

Observe further that, under Assumption 2, $g^*(\boldsymbol{y})$ is separable over its coordinates. As a consequence, the domain $\mathcal{Y}$ of g^* can be expressed as the Cartesian product of $\mathrm{dom}(g_i^*)$. This structure is crucial for variance reduction, as the algorithm in this case relies on performing coordinate descent updates over the dual variables $\boldsymbol{y}$.

Primal-dual gap. Given $\boldsymbol{x} \in \mathbb{R}^d$, the *primal value* of the problem (PD) is $P(\boldsymbol{x}) = \max_{\boldsymbol{v}\in\mathbb{R}^n} L(\boldsymbol{x}, \boldsymbol{v})$. Similarly, the *dual value* (PD) is defined by $D(\boldsymbol{y}) = \min_{\boldsymbol{u}\in\mathbb{R}^d} L(\boldsymbol{u}, \boldsymbol{y})$. Given a primal-dual pair $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^d \times \mathbb{R}^n$, primal-dual gap is then defined by $\mathrm{Gap}(\boldsymbol{x}, \boldsymbol{y}) = P(\boldsymbol{x}) - D(\boldsymbol{y}) = \max_{(\boldsymbol{u},\boldsymbol{v})\in\mathbb{R}^d\times\mathbb{R}^n} \mathrm{Gap}^{\boldsymbol{u},\boldsymbol{v}}(\boldsymbol{x}, \boldsymbol{y})$, where we define

$$\mathrm{Gap}^{\boldsymbol{u},\boldsymbol{v}}(\boldsymbol{x}, \boldsymbol{y}) = L(\boldsymbol{x}, \boldsymbol{v}) - L(\boldsymbol{u}, \boldsymbol{y}). \tag{1}$$

Observe that, by definition of $P(\boldsymbol{x})$ and $D(\boldsymbol{y})$, the maximum of $\mathrm{Gap}^{\boldsymbol{u},\boldsymbol{v}}(\boldsymbol{x},\boldsymbol{y})$ for fixed $(\boldsymbol{x},\boldsymbol{y})$ is attained when $(\boldsymbol{u},\boldsymbol{v}) \in \mathcal{X}\times\mathcal{Y}$, so we can also write $\mathrm{Gap}(\boldsymbol{x},\boldsymbol{y}) = \max_{(\boldsymbol{u},\boldsymbol{v})\in\mathcal{X}\times\mathcal{Y}} \mathrm{Gap}^{\boldsymbol{u},\boldsymbol{v}}(\boldsymbol{x},\boldsymbol{y})$.

For our analysis, it is useful to work with the relaxed gap $\mathrm{Gap}^{\boldsymbol{u},\boldsymbol{v}}(\boldsymbol{x},\boldsymbol{y})$. In particular, to bound the primal-dual gap $\mathrm{Gap}(\boldsymbol{x},\boldsymbol{y})$ for a candidate solution pair $(\boldsymbol{x},\boldsymbol{y})$ constructed by the algorithm, we first bound $\mathrm{Gap}^{\boldsymbol{u},\boldsymbol{v}}(\boldsymbol{x},\boldsymbol{y})$ for arbitrary $(\boldsymbol{u},\boldsymbol{v}) \in \mathcal{X}\times\mathcal{Y}$. The bound on $\mathrm{Gap}(\boldsymbol{x},\boldsymbol{y})$ then follows by taking the supremum of $\mathrm{Gap}^{\boldsymbol{u},\boldsymbol{v}}(\boldsymbol{x},\boldsymbol{y})$ over $(\boldsymbol{u},\boldsymbol{v}) \in \mathcal{X}\times\mathcal{Y}$. In general, $\mathrm{Gap}(\boldsymbol{x},\boldsymbol{y})$ can be bounded by a finite quantity only when $\mathcal{X},\mathcal{Y}$ are compact (Nesterov, 2005b; Ouyang & Xu, 2019). If either of $\mathcal{X},\mathcal{Y}$ is unbounded, to provide meaningful results and similar to Chambolle & Pock (2011), we assume that an optimal primal-dual pair $(\boldsymbol{x}^*,\boldsymbol{y}^*)$ for which $\mathrm{Gap}(\boldsymbol{x}^*,\boldsymbol{y}^*) = 0$ exists, and bound the primal-dual gap in a ball around $(\boldsymbol{x}^*,\boldsymbol{y}^*)$.

Auxiliary results. Additional auxiliary results on growth of sequences that are needed when establishing convergence rates in our results are provided in Appendix C.

3. Primal-Dual Accelerated Dual Averaging

In this section, we provide the PDA^2 algorithm for solving Problem (PD) under Assumption 1. The results in this section provide the basis for our results in Section 4 for the finite-sum primal-dual setting.

PDA^2 is described in Algorithm 1. Observe that the points $\boldsymbol{u}$, $\boldsymbol{v}$ in the definitions of estimate sequences $\phi_k(\boldsymbol{x}), \psi_k(\boldsymbol{y})$ do not play a role in the definitions of $\boldsymbol{x}_k$, $\boldsymbol{y}_k$, as the corresponding $\arg\min$s are independent of $\boldsymbol{u}$ and $\boldsymbol{v}$. They appear in the definitions of $\phi_k(\boldsymbol{x}), \psi_k(\boldsymbol{y})$ only for the convenience of the convergence analysis; the algorithm itself can be stated without them.

We now outline the main technical ideas in the PDA^2 algorithm. To bound the relaxed notion of the primal-dual gap $\mathrm{Gap}^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\boldsymbol{x}}_k,\tilde{\boldsymbol{y}}_k)$ discussed in Section 2, we use estimate sequences $\phi_k(\boldsymbol{x})$ and $\psi_k(\boldsymbol{y})$ defined in the algorithm. Unlike the classical estimate sequences used, for example, in Nesterov (2005b), these estimate sequences do not directly estimate the values of the primal and dual, but instead contain additional bilinear terms, which are crucial for forming an intricate coupling argument between the primal and the dual that leads to the desired convergence bounds. In particular, the bilinear term in the definition of ψ_k is defined w.r.t. an extrapolated point $\bar{\boldsymbol{x}}_{k-1}$. This extrapolated point is not guaranteed to lie in the domain of ℓ, but because this point appears only in bilinear terms, we never need to evaluate either ℓ or its subgradient at $\bar{\boldsymbol{x}}_{k-1}$. Instead, the extrapolated point plays a role in cancelling error terms that appear when relating the estimate sequences to $\mathrm{Gap}^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\boldsymbol{x}}_k,\tilde{\boldsymbol{y}}_k)$.

Our main technical result for this section concerning the convergence of PDA^2 is summarized in the following theorem. The proof of this result and supporting technical results are provided in Appendix A.

Algorithm 1 Primal-Dual Accelerated Dual Averaging (PDA^2)

Input: $(\boldsymbol{x}_0,\boldsymbol{y}_0) \in \mathcal{X}\times\mathcal{Y}, (\boldsymbol{u},\boldsymbol{v}) \in \mathcal{X}\times\mathcal{Y}, \sigma \geq 0, \gamma \geq 0, \|\boldsymbol{B}\| = R > 0, K$.
$a_0 = A_0 = 0$.
$\boldsymbol{x}_0 = \boldsymbol{x}_{-1} \in \mathbb{R}^d, \boldsymbol{y}_0 \in \mathbb{R}^n$.
$\phi_0(\cdot) = \frac{1}{2}\|\cdot - \boldsymbol{x}_0\|^2, \psi_0(\cdot) = \frac{1}{2}\|\cdot - \boldsymbol{y}_0\|^2$.
for $k = 1, 2, \ldots, K$ **do**
  $a_k = \frac{\sqrt{(1+\sigma A_{k-1})(1+\gamma A_{k-1})}}{\sqrt{2}R}, A_k = A_{k-1} + a_k$.
  $\bar{\boldsymbol{x}}_{k-1} = \boldsymbol{x}_{k-1} + \frac{a_{k-1}}{a_k}(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-2})$.
  $\boldsymbol{y}_k = \arg\min_{\boldsymbol{y}\in\mathbb{R}^n}\{\psi_k(\boldsymbol{y}) = \psi_{k-1}(\boldsymbol{y}) + a_k(\langle -\boldsymbol{B}\bar{\boldsymbol{x}}_{k-1}, \boldsymbol{y} - \boldsymbol{v}\rangle + g^*(\boldsymbol{y}))\}$.
  $\boldsymbol{x}_k = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d}\{\phi_k(\boldsymbol{x}) = \phi_{k-1}(\boldsymbol{x}) + a_k(\langle \boldsymbol{x} - \boldsymbol{u}, \boldsymbol{B}^T\boldsymbol{y}_k\rangle + \ell(\boldsymbol{x}))\}$.
end for
return $\tilde{\boldsymbol{y}}_K = \frac{1}{A_K}\sum_{k=1}^K a_k\boldsymbol{y}_k$, $\tilde{\boldsymbol{x}}_K = \frac{1}{A_K}\sum_{k=1}^K a_k\boldsymbol{x}_k$.

Theorem 1. *Under Assumption 1, for Algorithm 1, we have, $\forall(\boldsymbol{u},\boldsymbol{v}) \in \mathcal{X}\times\mathcal{Y}$ and $k \geq 1$,*

$$\mathrm{Gap}^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\boldsymbol{x}}_k,\tilde{\boldsymbol{y}}_k) \leq \frac{\|\boldsymbol{u}-\boldsymbol{x}_0\|^2 + \|\boldsymbol{v}-\boldsymbol{y}_0\|^2}{2A_k},$$

where $\tilde{\boldsymbol{x}}_k = \frac{1}{A_k}\sum_{i=1}^k a_i\boldsymbol{x}_i$, $\tilde{\boldsymbol{y}}_k = \frac{1}{A_k}\sum_{i=1}^k a_i\boldsymbol{y}_i$.

Further, if $(\boldsymbol{x}^*,\boldsymbol{y}^*)$ is a primal-dual solution to (PD), then

$$\begin{aligned}(1+\sigma A_k)\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 + \frac{1+\gamma A_k}{2}\|\boldsymbol{y}_k - \boldsymbol{y}^*\|^2 \\ \leq \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 + \|\boldsymbol{y}_0 - \boldsymbol{y}^*\|^2.\end{aligned} \tag{2}$$

In both cases, the growth of A_k can be bounded below as

$$\begin{aligned}A_k \geq \frac{1}{\sqrt{2}R}\max\Big\{&k,\ \Big(1+\frac{\sqrt{\sigma\gamma}}{\sqrt{2}R}\Big)^{k-1},\\ &\frac{\sigma}{9\sqrt{2}R}\Big([k-k_0]_+ + \max\{3.5\sqrt{R},\,1\}\Big)^2,\\ &\frac{\gamma}{9\sqrt{2}R}\Big([k-k_0']_+ + \max\{3.5\sqrt{R},\,1\}\Big)^2\Big\},\end{aligned}$$

where $[\cdot]_+ = \max\{\cdot, 0\}$, $k_0 = \lceil\frac{\sigma}{9\sqrt{2}R}\rceil$, and $k_0' = \lceil\frac{\gamma}{9\sqrt{2}R}\rceil$.

Remark 1. *As $\sigma \geq 0$ and $\gamma \geq 0$, Theorem 1 guarantees that all iterates of PDA^2 remain within a bounded set, due to Eq. (2). In particular, $\boldsymbol{x}_k \in \mathcal{B}(\boldsymbol{x}^*, r_0)$, $\boldsymbol{y}_k \in \mathcal{B}(\boldsymbol{y}^*, \sqrt{2}r_0)$, where $r_0 = \sqrt{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 + \|\boldsymbol{y}_0 - \boldsymbol{y}^*\|^2}$ and $\mathcal{B}(\boldsymbol{z}, r)$ denotes the Euclidean ball of radius r, centered at $\boldsymbol{z}$. Moreover, by rearranging Eq. (2), we can conclude that $\|\boldsymbol{x}^* - \boldsymbol{x}_k\|^2 \leq \frac{{r_0}^2}{1+\sigma A_k}$ and $\|\boldsymbol{y}^* - \boldsymbol{y}_k\|^2 \leq \frac{2{r_0}^2}{1+\gamma A_k}$.*

Remark 2. *Observe that when the domains of g^* and ℓ are bounded (i.e., when $D_{\mathcal{X}} < \infty$, $D_{\mathcal{Y}} < \infty$, and, in particular, in the setting of constrained optimization over compact sets), Theorem 1 implies the following bound on the primal-dual gap $\mathrm{Gap}(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{y}}_k) \leq \frac{{D_{\mathcal{X}}}^2 + {D_{\mathcal{Y}}}^2}{2A_k}$. This bound can be shown to be optimal, using results from Ouyang & Xu (2019). For unbounded domains of g^* and ℓ, it is generally not possible to have any finite bound on $\mathrm{Gap}(\boldsymbol{x}, \boldsymbol{y})$ unless $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^*, \boldsymbol{y}^*)$ (for a concrete example, see, e.g., Diakonikolas (2020)). In such a case, it is common to restrict $\boldsymbol{u}, \boldsymbol{v}$ to bounded sets that include $\boldsymbol{x}^*, \boldsymbol{y}^*$, such as $\mathcal{B}(\boldsymbol{x}^*, r_0)$, $\mathcal{B}(\boldsymbol{y}^*, \sqrt{2}r_0)$ from Remark 1 (Chambolle & Pock, 2011).*

Remark 3. *To bound the function value gap $f(\tilde{\boldsymbol{x}}_k) - f(\boldsymbol{x}^*)$ for Problem (P) using Theorem 1, we need only that $D_{\mathcal{Y}}$ is bounded, leading to the bound $f(\tilde{\boldsymbol{x}}_k) - f(\boldsymbol{x}^*) \leq \frac{4{r_0}^2 + {D_{\mathcal{Y}}}^2}{A_k}$, where $r_0 = \sqrt{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 + \|\boldsymbol{y}_0 - \boldsymbol{y}^*\|^2}$ as in Remark 1, since for $\boldsymbol{u} \in \mathcal{B}(\boldsymbol{x}^*, r_0)$ we have that $\|\boldsymbol{u} - \boldsymbol{x}_0\| \leq 2r_0$. To see this, note that, as the iterates $\boldsymbol{x}_i$ of PDA^2 are guaranteed to remain in $\mathcal{B}(\boldsymbol{x}^*, r_0)$ (by Remark 1), there is no difference between applying this algorithm to f or to $f + I_{\mathcal{B}(\boldsymbol{x}^*, r_0)}$, where $I_{\mathcal{B}(\boldsymbol{x}^*, r_0)}$ is the indicator function of $\mathcal{B}(\boldsymbol{x}^*, r_0)$. This allows us to restrict $\boldsymbol{u} \in \mathcal{B}(\boldsymbol{x}^*, r_0)$ when bounding $f(\tilde{\boldsymbol{x}}_k) - f(\boldsymbol{x}^*)$ by $\mathrm{Gap}^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{y}}_k)$. Note that for typical instances of nonsmooth ERM problems, the domain $\mathcal{Y}$ of g^* is compact. Further, if g^* is strongly convex ($\gamma > 0$), then the set $\tilde{\mathcal{Y}} := \{\arg\max_{\boldsymbol{y} \in \mathbb{R}^n} \langle \boldsymbol{B}\boldsymbol{x}, \boldsymbol{y}\rangle - g^*(\boldsymbol{y}) : \boldsymbol{x} \in \mathcal{B}(\boldsymbol{x}^*, r_0)\}$ is guaranteed to be compact. This claim follows from standard results, as in this case $\arg\max_{\boldsymbol{y} \in \mathbb{R}^n} \langle \boldsymbol{B}\boldsymbol{x}, \boldsymbol{y}\rangle - g^*(\boldsymbol{y}) = \nabla g(\boldsymbol{B}\boldsymbol{x})$ (by the standard Fenchel-Young inequality; see, e.g., Rockafellar & Wets (2009, Proposition 11.3)) and g is $\frac{1}{\gamma}$-smooth. Thus, $\sup_{\boldsymbol{v},\boldsymbol{y} \in \tilde{\mathcal{Y}}} \|\boldsymbol{v} - \boldsymbol{y}\| = \sup_{\boldsymbol{x},\boldsymbol{u} \in \mathcal{B}(\boldsymbol{x}^*, r_0)} \|\nabla g(\boldsymbol{B}\boldsymbol{x}) - \nabla g(\boldsymbol{B}\boldsymbol{u})\| \leq \frac{R}{\gamma} r_0$.*

4. Variance Reduction via Primal-Dual Accelerated Dual Averaging

We now study the finite-sum form (FS-PD) of (PD), making use of the properties of the finite-sum terms described in Assumption 2. In Algorithm 2, we describe VRPDA^2 which is a randomized coordinate variant of the PDA^2 algorithm from Section 3. By extending the unified nature of PDA^2, VRPDA^2 provides a unified and simplified treatment for both the general convex-general concave ($\sigma = 0$) setting and the strongly convex-general concave setting.

To provide an algorithm with complexity better than the deterministic counterpart PDA^2, we combine the deterministic initialization strategy of full primal-dual update in Steps 4-6 with randomized primal-dual updates in the main loop—a strategy inspired by the recent paper of Song et al. (2020a). The use of the factor n during initialization, in Step 7, helps to cancel an error term of order $O(n)$ in the analysis.

The main loop (Steps 8-15) randomizes the main loop of PDA^2 by introducing sampling in Step 10 and adding an auxiliary variable $\boldsymbol{z}_k$ that is updated with $O(d)$ cost in Step 13. ($\boldsymbol{z}_1$ is initialized in Step 5.) In Step 11, we update the estimate sequence ψ_k by adding a term involving only the j_k component of the finite sum, rather than the entire sum, as is required in Step 8 of Algorithm 1. As a result, although we define the estimate sequence for the entire vector $\boldsymbol{y}_k$, each update to $\boldsymbol{y}_k$ requires updating *only the j_k coordinate* of $\boldsymbol{y}_k$. In Step 12, we use a "variance reduced gradient" $\boldsymbol{z}_{k-1} + (y_{k,j_k} - y_{k-1,j_k})\boldsymbol{b}_{j_k}$ to update ϕ_k, helping to cancel the error from the randomized update of Step 11. The update of the sequences $\{a_k\}$, $\{A_k\}$ appears at the end of the main loop, to accommodate their modified definitions. The modified update for a_{k+1} ensures that a_k cannot have exponential growth with a rate higher than $\left(1 + \frac{1}{n-1}\right)$, which is an intrinsic constraint for sampling with replacement (see Song et al. (2020a); Hannah et al. (2018)).

Finally, as Algorithm 2 is tailored to the nonsmooth ERM problem (P), we only return the last iterate $\boldsymbol{x}_k$ or the weighed average iterate $\tilde{\boldsymbol{x}}_k$ on the primal side, even though we provide guarantees for both primal and dual variables.

Algorithm 2 provides sufficient detail for the convergence analysis, but its efficient implementation is not immediately clear, due especially to Step 11. An implementable version is described in Appendix D, showing that the per-iteration cost is $O(d)$ and that $O(n)$ additional storage is required.

Our main technical result is summarized in Theorem 2. Its proof relies on three main technical lemmas that bound the growth of estimate sequences $\phi_k(\boldsymbol{x}_k)$ and $\psi_k(\boldsymbol{y}_k)$ below and above. Proofs are provided in Appendices B and C.

Theorem 2. *Suppose that Assumption 2 holds. Then for any $(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{X} \times \mathcal{Y}$, the vectors $\boldsymbol{x}_k$, $\boldsymbol{y}_k$, $k = 2, 3, \ldots, K$ and the average $\tilde{\boldsymbol{x}}_k = \frac{1}{A_k}\sum_{i=1}^k a_i \boldsymbol{x}_i$ generated by Algorithm 2 satisfy the following bound for $k = 2, 3, \ldots, K$:*

$$\mathbb{E}[\mathrm{Gap}^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{y}}_k)] \leq \frac{n(\|\boldsymbol{u} - \boldsymbol{x}_0\|^2 + \|\boldsymbol{v} - \boldsymbol{y}_0\|^2)}{2A_k},$$

where $\tilde{\boldsymbol{y}}_k := \frac{na_k\boldsymbol{y}_k + \sum_{i=2}^{k-1}(na_i - (n-1)a_{i+1})\boldsymbol{y}_i}{A_k}$.

Moreover, if $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is a primal-dual solution to (PD), then

$$\mathbb{E}\Big[\frac{n}{4}\|\boldsymbol{y}^* - \boldsymbol{y}_k\|^2 + \frac{n + \sigma A_k}{2}\|\boldsymbol{x}^* - \boldsymbol{x}_k\|^2\Big] \leq \frac{n(\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2 + \|\boldsymbol{y}^* - \boldsymbol{y}_0\|^2)}{2}.$$

In both cases, A_k is bounded below as follows:

$$A_k \geq \max\Big\{\frac{n-1}{2R'}\Big(1 + \frac{1}{n-1}\Big)^k \mathbb{1}_{k \leq k_0}, \frac{(n-1)^2\sigma}{(4R')^2 n}(k - k_0 + n - 1)^2 \mathbb{1}_{k \geq k_0}, \frac{n(k - K_0 + n - 1)}{2R'}\mathbb{1}_{k \geq K_0}\Big\},$$

Algorithm 2 Variance Reduction via Primal-Dual Accelerated Dual Averaging (VRPDA^2)

Input: $(\boldsymbol{x}_0, \boldsymbol{y}_0) \in \mathcal{X} \times \mathcal{Y}, (\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{X} \times \mathcal{Y}, \sigma \geq 0, R' > 0, K, n.$
$\phi_0(\cdot) = \frac{1}{2}\|\cdot - \boldsymbol{x}_0\|^2, \psi_0(\cdot) = \frac{1}{2}\|\cdot - \boldsymbol{y}_0\|^2.$
$a_0 = A_0 = 0, \tilde{a}_1 = \frac{1}{2R'}.$
$\boldsymbol{y}_1 = \arg\min_{\boldsymbol{y}\in\mathbb{R}^n}\{\tilde{\psi}_1(\boldsymbol{y}) := \psi_0(\boldsymbol{y}) + \tilde{a}_1(\langle -\boldsymbol{B}\boldsymbol{x}_0, \boldsymbol{y} - \boldsymbol{v}\rangle + g^*(\boldsymbol{y}))\}.$
$\boldsymbol{z}_1 = \boldsymbol{B}^T\boldsymbol{y}_1.$
$\boldsymbol{x}_1 = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d}\{\tilde{\phi}_1(\boldsymbol{x}) := \phi_0(\boldsymbol{x}) + \tilde{a}_1(\langle \boldsymbol{x} - \boldsymbol{u}, \boldsymbol{z}_1\rangle + \ell(\boldsymbol{x}))\}.$
$\psi_1 := n\tilde{\psi}_1, \phi_1 := n\tilde{\phi}_1, a_1 = A_1 = n\tilde{a}_1, a_2 = \frac{1}{n-1}a_1, A_2 = A_1 + a_2.$
for $k = 2, 3, \ldots, K$ **do**
  $\bar{\boldsymbol{x}}_{k-1} = \boldsymbol{x}_{k-1} + \frac{a_{k-1}}{a_k}(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-2}).$
  Pick j_k uniformly at random in $[n]$.
  $\boldsymbol{y}_k = \arg\min_{\boldsymbol{y}\in\mathbb{R}^n}\{\psi_k(\boldsymbol{y}) = \psi_{k-1}(\boldsymbol{y}) + a_k(-\boldsymbol{b}_{j_k}^T\bar{\boldsymbol{x}}_{k-1}(y_{j_k} - v_{j_k}) + g_{j_k}^*(y_{j_k}))\}.$
  $\boldsymbol{x}_k = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d}\{\phi_k(\boldsymbol{x}) = \phi_{k-1}(\boldsymbol{x}) + a_k(\langle \boldsymbol{x} - \boldsymbol{u}, \boldsymbol{z}_{k-1} + (y_{k,j_k} - y_{k-1,j_k})\boldsymbol{b}_{j_k}\rangle + \ell(\boldsymbol{x}))\}.$
  $\boldsymbol{z}_k = \boldsymbol{z}_{k-1} + \frac{1}{n}(y_{k,j_k} - y_{k-1,j_k})\boldsymbol{b}_{j_k}.$
  $a_{k+1} = \min\left(\left(1 + \frac{1}{n-1}\right)a_k, \frac{\sqrt{n(n+\sigma A_k)}}{2R'}\right), A_{k+1} = A_k + a_{k+1}.$
end for
return $\boldsymbol{x}_K$ or $\tilde{\boldsymbol{x}}_K := \frac{1}{A_K}\sum_{i=1}^K a_i\boldsymbol{x}_i.$

where $\mathbb{1}$ *denotes the indicator function,* $K_0 = \lceil \frac{\log(n)}{\log(n)-\log(n-1)} \rceil$, $k_0 = \lceil \frac{\log B_{n,\sigma,R'}}{\log(n)-\log(n-1)} \rceil$, *and*

$$B_{n,\sigma,R'} = \frac{\sigma n(n-1)}{4R'} + \sqrt{\left(\frac{\sigma n(n-1)}{4R'}\right)^2 + n^2} \geq n\max\left\{1, \frac{\sigma(n-1)}{2R'}\right\}.$$

Observe that, due to the randomized nature of the algorithm, the convergence bounds are obtained in expectation w.r.t. the random choices of coordinates j_k over iterations.

Now let us comment on the iteration complexity of VRPDA^2, given target error $\epsilon > 0$. For concreteness, let $D^2 := \|\boldsymbol{u} - \boldsymbol{x}_0\|^2 + \|\boldsymbol{v} - \boldsymbol{y}_0\|^2$, where D^2 can be bounded using the same reasoning as in Remarks 2 and 3. To bound the gap by ϵ, we need $A_k \geq \frac{nD^2}{2\epsilon}$. When $\epsilon \geq \frac{nR'D^2}{(n-1)B_{n,R',\sigma}}$, then $k = \left\lceil \frac{\log(\frac{nR'D^2}{(n-1)\epsilon})}{\log(n)-\log(n-1)} \right\rceil = O\big(n\log(\frac{R'D}{\epsilon})\big)$ iterations suffice, as in this case $k \leq k_0$. When $\epsilon < \frac{nR'D^2}{(n-1)B_{n,\sigma,R'}}$, then the bound on k is obtained by ensuring that either of the last two terms bounding A_k below in Theorem 2 is bounded below by $\frac{nD^2}{2\epsilon}$, leading to $k = O\big(n\log(B_{n,\sigma,R'}) + \min\{\frac{R'D}{\sqrt{\sigma\epsilon}}, \frac{R'D^2}{\epsilon}\}\big)$.

5. Numerical Experiments

We study the performance of VRPDA^2 using the elastic-net-regularized support vector machine (SVM) problem, which corresponds to (P) with $g_i(\boldsymbol{b}_i^T\boldsymbol{x}) = \max\{1 - c_i\boldsymbol{b}_i^T\boldsymbol{x}, 0\}$, $c_i \in \{1, -1\}$ and $\ell(\boldsymbol{x}) = \lambda\|\boldsymbol{x}\|_1 + \frac{\sigma}{2}\|\boldsymbol{x}\|_2^2$, $\lambda \geq 0$, $\sigma \geq 0$. This problem is nonsmooth and general convex if $\sigma = 0$ or strongly convex if $\sigma > 0$. Its primal-dual formulation is

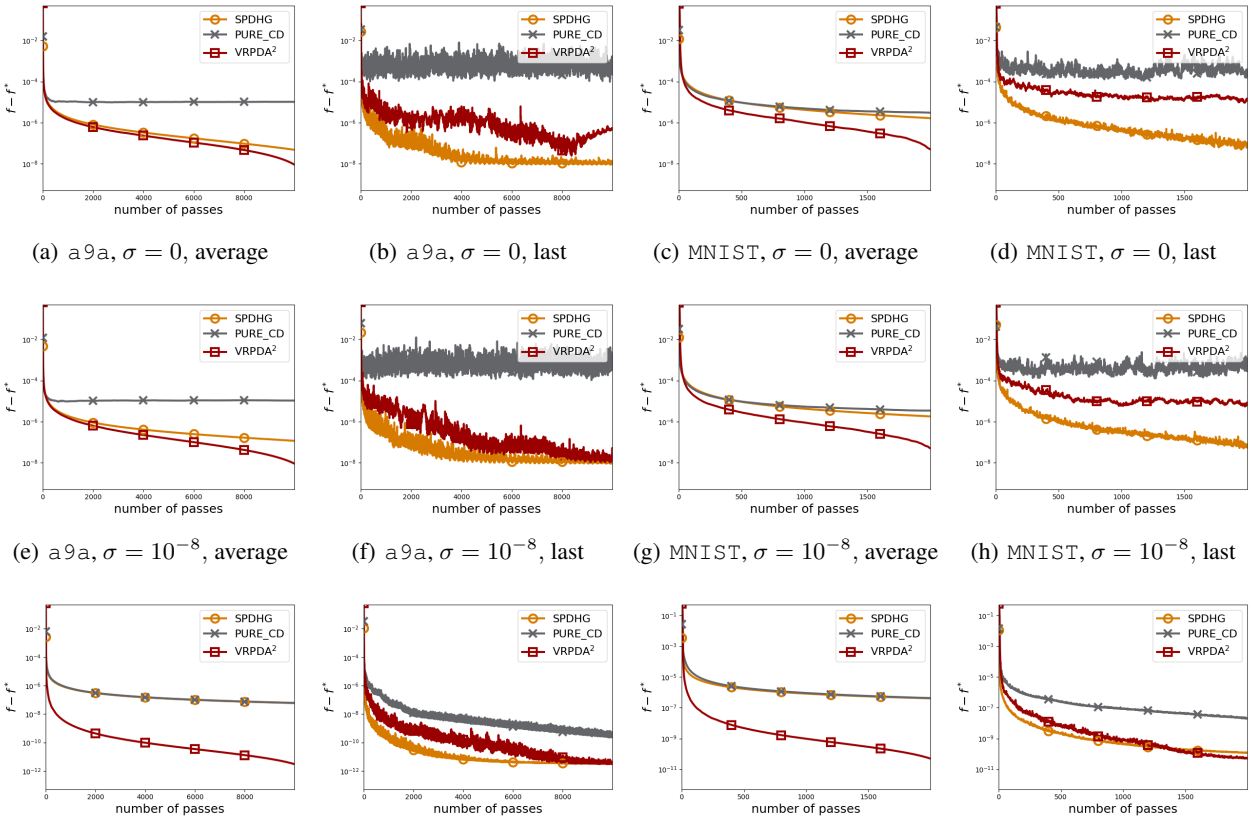
$$\min_{\boldsymbol{x}\in\mathbb{R}^d}\max_{-1\leq y_i\leq 0,\, i\in[n]} L(\boldsymbol{x}, \boldsymbol{y}),$$
$$L(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^n y_i\left(\langle c_i\boldsymbol{b}_i, \boldsymbol{x}\rangle - 1\right) + \lambda\|\boldsymbol{x}\|_1 + \frac{\sigma}{2}\|\boldsymbol{x}\|_2^2.$$

We compare VRPDA^2 with two competitive algorithms SPDHG (Chambolle et al., 2018) and PURE_CD (Alacaoglu et al., 2020) on standard `a9a` and `MNIST` datasets from the LIBSVM library (LIB).[1] Both datasets are large, with $n = 32,561$, $d = 123$ for `a9a`, and $n = 60,000$, $d = 780$ for `MNIST`. For simplicity, we normalize each data sample to unit Euclidean norm, so that the Lipschitz constants appearing in the analysis (such as R' in VRPDA^2) are at most 1. We then scale these Lipschitz constants by $\{0.1, 0.25, 0.5, 0.75, 1\}$[2]. As is standard for ERM, we plot the function value gap of the primal problem (P) in terms of the number of passes over the dataset. The plotted function value gap was evaluated using an estimated value $\tilde{f}^*$ of $f^* = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$. For the plots to depict an accurate estimate of the function value gap, the true function value gap $f - f^*$ must dominate the error of the estimate $\tilde{f}^* - f^*$. In our numerical experiments, this is achieved by running the algorithms 30 times as many iterations as are is shown in the plots, choosing the lowest function value $f_{\min}$ observed over this extended run and over all algorithms, and setting $\tilde{f}^* = f_{\min} - \delta$, where δ is either 10^{-8} or 10^{-13}, depending on the value of σ.

We fix the ℓ_1-regularization parameter λ to 10^{-4} and vary $\sigma \in \{0, 10^{-8}, 10^{-4}\}$, to represent the general convex, ill-

[1] For each sample of `MNIST`, we reassign the label as 1 if it is in $\{5, 6, \ldots, 9\}$ and -1 otherwise.

[2] In our experiments, all the algorithms diverge when the Lipschitz constant is set to 0.1.



(a) a9a, $\sigma = 0$, average (b) a9a, $\sigma = 0$, last (c) MNIST, $\sigma = 0$, average (d) MNIST, $\sigma = 0$, last

(e) a9a, $\sigma = 10^{-8}$, average (f) a9a, $\sigma = 10^{-8}$, last (g) MNIST, $\sigma = 10^{-8}$, average (h) MNIST, $\sigma = 10^{-8}$, last

(i) a9a, $\sigma = 10^{-4}$, average (j) a9a, $\sigma = 10^{-4}$, last (k) MNIST, $\sigma = 10^{-4}$, average (l) MNIST, $\sigma = 10^{-4}$, last

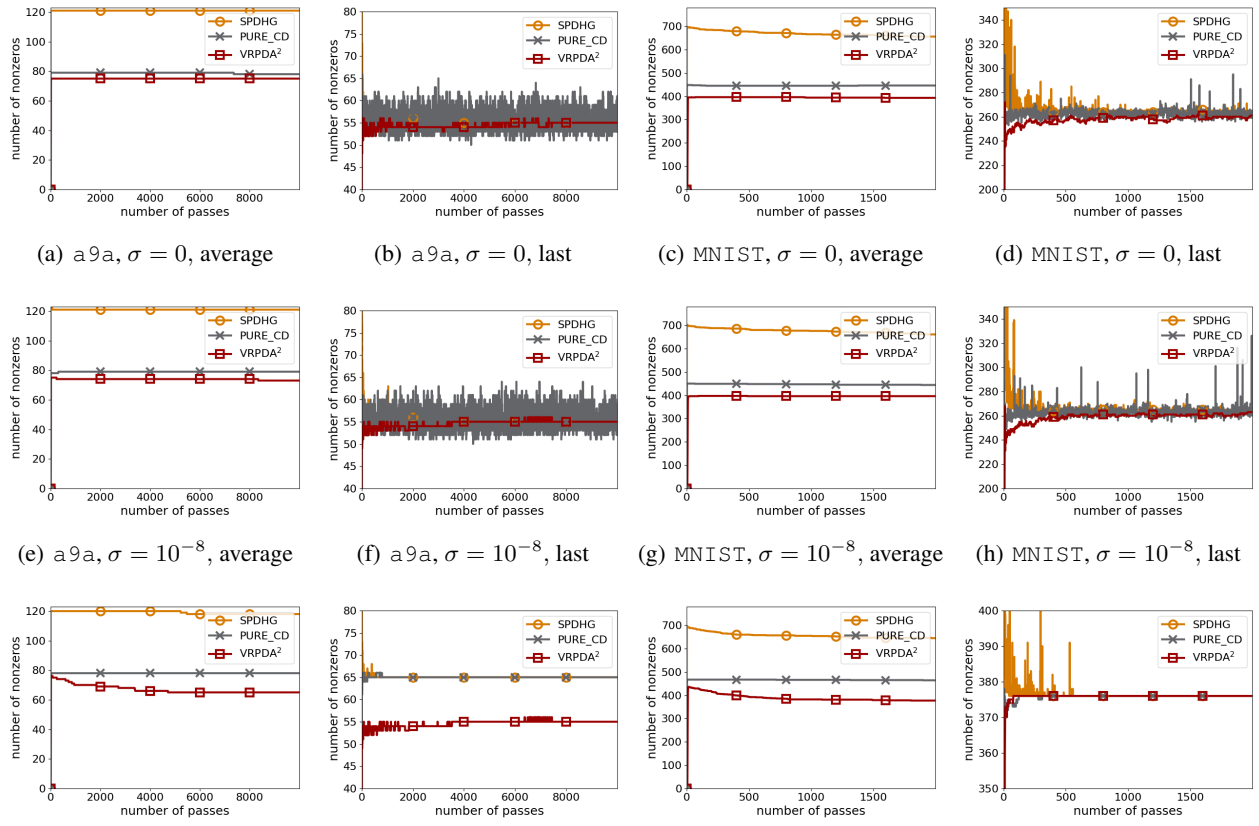
Figure 1. Comparison of VRPDA^2 to SPDHG and PURE_CD run for the elastic net-regularized SVM, on a9a and MNIST datasets. In all the plots, σ is the strong convexity parameter of the regularizer ℓ; "last" refers to the last iterate, "average" to the average iterate. For all problem instances, VRPDA^2 attains either similar or improved convergence compared to other algorithms.

conditioned strongly convex, and well-conditioned strongly convex settings, respectively. For all the settings, we provide the comparison in terms of the average and last iterate[3]. As can be observed from Figure 1, the iterate averaging yields much smoother curves, decreasing monotonically, and is generally more accurate than the last iterate. This is expected for the nonsmooth and general convex setting, as there are no theoretical guarantees for the last iterate, while for other cases the guarantee for the last iterate is on the distance to optimum, not the primal gap. As can be seen in Figure 1, the average iterate of VRPDA^2 is either competitive with or improves upon SPDHG and PURE_CD.

As can be observed from Figure 1, there is a noticeable difference in the performance of all the algorithms when their function value gap is evaluated at the average iterate versus the last iterate. For VRPDA^2 , a dual averaging-style method that has a sparsity-promoting property (Xiao, 2010), this difference comes from the significantly different sparsity of the average iterate and last iterate. As shown in Figure 2, the average iterate is less sparse but provides a more accurate fit, while the last iterate is sparser (and thus more robust) but less accurate. For SPDHG, the last iterate is significantly more accurate than the average iterate in the strongly convex settings ($\sigma \in \{10^{-8}, 10^{-4}\}$), because simple uniform average we use may not be the best choice for the two settings. Meanwhile, the better performance of SPDHG compared with VRPDA^2 in terms of the last iterate is due partly to the fact that it is a mirror descent-style algorithm with less-sparse last iterate. In our experiments, the PURE_CD algorithm is always worse than VRPDA^2 and SPDHG, which is partly consistent with its worse convergence guarantee as shown in Table 1. However, as PURE_CD is targeted to sparse datasets, it may have a better runtime performance in such settings, as shown in Alacaoglu et al. (2020).

Meanwhile, the performance of the average iterate of VRPDA^2 and the last iterate of SPDHG is almost the same (the figures for the average iterate and the last iterate under the

[3] SPDHG and PURE_CD provide no results for the average iterate in the nonsmooth and strongly convex setting, so we use simple uniform average for both.



(a) a9a, $\sigma = 0$, average (b) a9a, $\sigma = 0$, last (c) MNIST, $\sigma = 0$, average (d) MNIST, $\sigma = 0$, last

(e) a9a, $\sigma = 10^{-8}$, average (f) a9a, $\sigma = 10^{-8}$, last (g) MNIST, $\sigma = 10^{-8}$, average (h) MNIST, $\sigma = 10^{-8}$, last

(i) a9a, $\sigma = 10^{-4}$, average (j) a9a, $\sigma = 10^{-4}$, last (k) MNIST, $\sigma = 10^{-4}$, average (l) MNIST, $\sigma = 10^{-4}$, last

Figure 2. Comparison of sparsity for VRPDA^2, SPDHG, and PURE_CD run for the elastic net-regularized SVM problem, on a9a and MNIST datasets. In all the plots, σ is the strong convexity parameter of the regularizer ℓ; "last" refers to the last iterate, "average" to the average iterate. For all problem instances, VRPDA^2 generally constructs the sparsest solutions out of the three algorithms. (The number of nonzeros is computed by counting the elements with absolute value larger than 10^{-7}.)

same setting use the same scale), which is surprising in that VRPDA^2 has n-times better theoretical guarantees than SPDHG for small ϵ. The better theoretical guarantee of VRPDA^2 comes from the particular initialization strategy inspired by Song et al. (2020a). Nevertheless, similar to the experimental results in Song et al. (2020a), no significant performance gain (or loss) due to this initialization strategy is observed in practice. Thus, it is of interest to explore whether the initialization strategy is essential for improved algorithm performance or if it is needed only for the theoretical argument to go through.

6. Discussion

We introduced VRPDA^2, a variance-reduced primal-dual accelerated dual averaging algorithm for structured nonsmooth ERM problems in machine learning. We show that VRPDA^2 leverages the separable structure of common ERM problems to achieves the best known convergence rates on this class of problems, with good practical performance. It even improves upon the lower bounds for (general, nonstructured) composite optimization. It remain an open question to obtain tighter lower bounds for the problem class to which VRPDA^2 applies, possibly certifying its optimality, at least for small target error ϵ. Another direction is addressing settings with strongly convex loss functions currently not addressed by VRPDA^2, which may require very different techniques.

Acknowledgements

CS acknowledges support from the NSF award 2023239. JD acknowledges support from the Office of the Vice Chancellor for Research and Graduate Education at the University of Wisconsin–Madison with funding from the Wisconsin Alumni Research Foundation. SW acknowledges support from NSF Awards 1740707, 1839338, 1934612, and 2023239 and Subcontract 8F-30039 from Argonne National Laboratory.

References

LIBSVM Library. https://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html. Accessed: Feb. 3, 2020.

Alacaoglu, A., Dinh, Q. T., Fercoq, O., and Cevher, V. Smooth primal-dual coordinate descent algorithms for nonsmooth convex optimization. In *Proc. NIPS'17*, 2017.

Alacaoglu, A., Fercoq, O., and Cevher, V. Random extrapolation for primal-dual coordinate descent. In *Proc. ICML'20*, 2020.

Allen-Zhu, Z. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proc. ACM STOC'17*, 2017.

Allen-Zhu, Z. and Hazan, E. Optimal black-box reductions between optimization objectives. In *Proc. NIPS'16*, 2016.

Allen-Zhu, Z. and Orecchia, L. Linear coupling: An ultimate unification of gradient and mirror descent. In *Proc. ITCS'17*, 2017.

Carmon, Y., Jin, Y., Sidford, A., and Tian, K. Variance reduction for matrix games. In *Proc. NeurIPS'19*, 2019.

Chambolle, A. and Pock, T. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

Chambolle, A., Ehrhardt, M. J., Richtárik, P., and Schonlieb, C.-B. Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *SIAM Journal on Optimization*, 28(4):2783–2808, 2018.

Chen, Y., Lan, G., and Ouyang, Y. Accelerated schemes for a class of variational inequalities. *Mathematical Programming*, 165(1):113–149, 2017.

Dang, C. and Lan, G. Randomized first-order methods for saddle point optimization. *arXiv preprint arXiv:1409.8625*, 2014.

Defazio, A., Bach, F., and Lacoste-Julien, S. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Proc. NIPS'14*, 2014.

Devraj, A. M. and Chen, J. Stochastic variance reduced primal dual algorithms for empirical composition optimization. In *Proc. NeurIPS'19*, 2019.

Diakonikolas, J. Halpern iteration for near-optimal and parameter-free monotone inclusion and strong solutions to variational inequalities. In *Proc. COLT'20*, 2020.

Diakonikolas, J., Daskalakis, C., and Jordan, M. I. Efficient methods for structured nonconvex-nonconcave min-max optimization. *arXiv preprint arXiv:2011.00364*, 2020.

Fercoq, O. and Bianchi, P. A coordinate-descent primal-dual algorithm with large step size and possibly nonseparable functions. *SIAM Journal on Optimization*, 29(1):100–134, 2019.

Hannah, R., Liu, Y., O'Connor, D., and Yin, W. Breaking the span assumption yields fast finite-sum minimization. In *Proc. NeurIPS'18*, 2018.

Hazan, E. et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Proc. NIPS'13*, 2013.

Lan, G., Li, Z., and Zhou, Y. A unified variance-reduced accelerated gradient method for convex optimization. In *Proc. NeurIPS'19*, 2019.

Latafat, P., Freris, N. M., and Patrinos, P. A new randomized block-coordinate primal-dual proximal algorithm for distributed optimization. *IEEE Transactions on Automatic Control*, 64(10):4050–4065, 2019.

Lei, Q., Zhuo, J., Caramanis, C., Dhillon, I. S., and Dimakis, A. G. Primal-dual block generalized Frank-Wolfe. In *Proc. NeurIPS'19*, 2019.

Lin, Q., Lu, Z., and Xiao, L. An accelerated proximal coordinate gradient method. In *Proc. NIPS'14*, 2014.

Nemirovski, A. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

Nesterov, Y. Excessive gap technique in nonsmooth convex minimization. *SIAM Journal on Optimization*, 16(1): 235–249, 2005a.

Nesterov, Y. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005b.

Nesterov, Y. Dual extrapolation and its applications to solving variational inequalities and related problems. *Mathematical Programming*, 109(2-3):319–344, 2007.

Nesterov, Y. Universal gradient methods for convex optimization problems. *Mathematical Programming*, 152 (1-2):381–404, 2015.

Ouyang, Y. and Xu, Y. Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems. *Mathematical Programming*, pp. 1–35, 2019.

Rockafellar, R. T. and Wets, R. J.-B. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

Roux, N., Schmidt, M., and Bach, F. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Proc. NIPS'12*, 2012.

Song, C., Jiang, Y., and Ma, Y. Variance reduction via accelerated dual averaging for finite-sum optimization. In *Proc. NeurIPS'20*, 2020a.

Song, C., Zhou, Z., Zhou, Y., Jiang, Y., and Ma, Y. Optimistic dual extrapolation for coherent non-monotone variational inequalities. In *Proc. NeurIPS'20*, 2020b.

Tan, C., Zhang, T., Ma, S., and Liu, J. Stochastic primal-dual method for empirical risk minimization with $O(1)$ per-iteration complexity. In *Proc. NeurIPS'18*, 2018.

Tran-Dinh, Q., Fercoq, O., and Cevher, V. A smooth primal-dual optimization framework for nonsmooth composite convex minimization. *SIAM Journal on Optimization*, 28 (1):96–134, 2018.

Woodworth, B. E. and Srebro, N. Tight complexity bounds for optimizing composite objectives. In *Proc. NIPS'16*, 2016.

Xiao, L. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.

Zhang, Y. and Lin, X. Stochastic primal-dual coordinate method for regularized empirical risk minimization. In *Proc. ICML'15*, 2015.

Zhou, K., Shang, F., and Cheng, J. A simple stochastic variance reduced algorithm with fast convergence rates. In *Proc. ICML'18*, 2018.