OPT2020: 12th Annual Workshop on Optimization for Machine Learning

Variance Reduction on Adaptive Stochastic Mirror Descent

Wenjie Li LI3549@PURDUE.EDU
Zhanyu Wang WANG4094@PURDUE.EDU
Yichen Zhang ZHANG@PURDUE.EDU
Guang Cheng CHENGG@PURDUE.EDU
Purdue University, West Lafayette, IN

Abstract

We study the application of the variance reduction technique on general adaptive stochastic mirror descent algorithms in nonsmooth nonconvex optimization problems. We prove that variance reduction helps to reduce the gradient complexity of most general stochastic mirror descent algorithms, so it works well with time-varying steps sizes and adaptive optimization algorithms such as AdaGrad. We check the validity of our claims using experiments in deep learning.

1. Introduction

In this work, we study the non-smooth non-convex finite sum problem

$$\min_{x\in\mathcal{X}} F(x) := f(x) + h(x)$$

where $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$ and each f_i is a smooth but possibly non-convex function, and $h(x)$ is a non-smooth convex function, for example, L_1 regularization. Recently, the smooth version of the problem has been thoroughly studied, i.e. when $h(x) = 0$. To make the convergence of stochastic gradient descent (SGD) methods faster in such cases, the famous Stochastic Variance Reduced Gradient method (SVRG) [12] and its popular variants were proposed, such as SAGA [5], SCSG [17], SNVRG [27], SPIDER [8], stablized SVRG [9], and Natasha momentum variants [1, 2].

When it comes to the non-smooth case, a few algorithms based on the mirror descent algorithm [3, 6] have been proposed recently. For example, Ghadimi et al. [10] proved the convergence rate of Proximal GD (ProxGD), Proximal SGD(ProxSGD), and Stochastic Mirror Descent (SMD) when the sample size is sufficiently large. Reddi et al. [22] showed the convergence of ProxSVRG and ProxSAGA, which were the proximal variants of SVRG and SAGA respectively. Li and Li [18] created ProxSVRG+ and obtained even faster convergence than ProxSVRG. However, all of the above extensions do not consider the case when the algorithm becomes adaptive, i.e., when the step sizes are not fixed or even when the proximal functions in mirror descent are not fixed.

Instead of trying to create even faster algorithms in the nonsmooth setting, this work focuses on answering a more general question: Can the variance reduction technique accelerate the convergence of the general adaptive mirror descent algorithm? We give an affirmative answer to this question, as long as strong convexity of the proximal function is lower bounded by a constant m.

Our Contributions. In this paper, we prove that the variance reduced general adaptive SMD algorithms can reduce the gradient complexity of the original algorithms, so variance reduction indeed can make them converge faster. Moreover, our theory implies many useful results. For example, time-varying step sizes are allowed for ProxSVRG+ (and many other algorithms), as long as the step

sizes are upper bounded by a constant $1/L$. Besides, our analysis provides a general intuition that larger batch sizes are needed when using variance reduction on adaptive SMD algorithms with weaker convexity. A very important by-product of our analysis is the conclusion that variance reduction works well with adaptive algorithms, such as AdaGrad [7] and RMSProp [24]. We examine the correctness of our claims carefully on the CIFAR-10 [15] and MNIST [23] datasets.

Notations. For two matrices A, B, we use $A \succeq B$ to denote that the matrix $A - B$ is positive semi-definite. For two real integers a, b, we use $a \wedge b, a \vee b$ as short-hands for $\min(a, b)$ and $\max(a, b)$. We use $\lfloor a \rfloor$ to denote the largest integer that is smaller than a. We use $\tilde{O}(\cdot)$ to hide logarithm factors in big-O notations. Moreover, we frequently use the notation $[n]$ to represent the set $\{1, 2, \cdots, n\}$.

2. Preliminaries

We present the preliminary assumptions used throughout this paper. We first recall the general SMD algorithm with adaptive *proximal functions* $\psi_t(x)$, where α_t is the step size, $g_t = \nabla f_{\mathcal{I}_j}(x_t)$ is the gradient from a random sample $\mathcal{I}_j$, and $h(x)$ is the regularization on the dual space.

$$x_{t+1} = \text{argmin}_x \{\alpha_t \langle g_t, x\rangle + \alpha_t h(x) + B_{\psi_t}(x, x_t)\} \tag{1}$$

where $B_{\psi_t}(x, x_t)$ is the Bregman divergence, defined as $B_{\psi_t}(x, y) = \psi_t(x) - \psi_t(y) - \langle \nabla \psi_t(y), x - y\rangle$. One special example of the above definition is the Euclidean distance $\frac{1}{2}\|x - y\|_2^2$ in ProxSGD, which is generated by $\psi_t(x) = \frac{1}{2}\|x\|_2^2$. In this work, we consider adaptive SMD algorithms whose proximal functions are all m-strongly convex (**A1**) for some real constant $m > 0$,

(**A1.**) The proximal functions $\psi_t(x)$ are all m-strongly convex with respect to $\|\cdot\|_2$, i.e.

$$\psi_t(y) \geq \psi_t(x) + \langle \nabla \psi_t(x), y - x\rangle + \frac{m}{2}\|y - x\|_2^2, \forall t > 0$$

The constant m can be viewed as a lower bound of the strong convexity of all the proximal functions $\{\psi_t(x)\}$ and therefore assumption **A1** is very weak. For example, if $\psi_t(x) = \phi_t(x) + \frac{c}{2}\|x\|_2^2, c > 0$, where each $\phi_t(x)$ is an arbitrary convex function, then $m = c$. If $\psi_t(x) = \frac{1}{2}\langle x, H_t x\rangle$, $H_t \in \mathbb{R}^{d\times d}$ and $H_t \succeq mI$, the algorithm covers all the adaptive optimizers with constant m added to the denominator to avoid division by zero. For the functions $\{f_i\}_{i=1}^n$, we assume the L-smoothness and bounded variance gradients conditions, which are standard in non-convex optimization analysis.

(**A2.**) Each function f_i is L-smooth, i.e.

$$\|\nabla f_i(x) - \nabla f_i(y)\|_2 \leq L\|x - y\|_2$$

(**A3.**) $f(x)$ have unbiased stochastic gradients with bounded variance σ^2, i.e.

$$\mathbb{E}_{i\sim[n]}[\nabla f_i(x)] = \nabla f(x), \qquad \mathbb{E}_{i\sim[n]}\|\nabla f_i(x) - \nabla f(x)\|_2^2 \leq \sigma^2$$

The convergence of algorithms in non-convex optimization problems is usually measured by the stationarity of the gradient $\nabla f(x)$, i.e. $\mathbb{E}[\|\nabla f(x)\|^2] \leq \epsilon^2$. However, due to the existence of $h(x)$ in the non-smooth setting, such a definition is no longer intuitive. Instead, we follow Li and Li [18] to use the definition of generalized gradient and the related convergence criterion. Given the generated parameters x_t by the algorithm, we define the generalized gradient at iteration t as

$$g_{X,t} = \frac{1}{\alpha_t}(x_t - x_{t+1}^+), \text{ where } x_{t+1}^+ = \text{argmin}_x \{\alpha_t \langle \nabla f(x_t), x\rangle + \alpha_t h(x) + B_{\psi_t}(x, x_t)\}$$

Correspondingly, the convergence criterion is the stationarity of the generalized gradient $\mathbb{E}[\|g_{X,t^*}\|^2] \leq \epsilon^2$. We use the stochastic first-order oracle (SFO) complexity to compare the convergence of different algorithms. When given the parameters x, SFO returns one stochastic gradient $\nabla f_i(x)$. The gradient complexity of general adaptive SMD is similar to that of non-adaptive SMD algorithm [10], i.e.

$$O(\frac{n}{\epsilon^2} \wedge \frac{\sigma^2}{\epsilon^4} + n \wedge \frac{\sigma^2}{\epsilon^2}) \tag{2}$$

The proof of this bound is provided in Appendix A for completeness

3. Algorithm and Convergence

3.1. Convergence of Adaptive SMD with Variance Reduction

We first present the variance reduced adaptive SMD algorithm, which is an extension of ProxSVRG+ [18]. The details are presented in Algorithm 1. Similar to the aforementioned paper, B_t and b_t are called the batch sizes and mini-batch sizes. The major difference between Algorithm 1 and ProxSVRG+ is that the proximal function is a general $\psi_{tk}(x)$ instead of the fixed $\psi(x) = \frac{1}{2}x^2$, and therefore naturally the Euclidean distance $\frac{1}{2}\|y - y_k^t\|_2^2$ is replaced by the general Bregman divergence $B_{\psi_{tk}}(y, y_k^t)$, so Algorithm 1 covers the ProxSVRG+ and a lot more algorithms. Now we present the major convergence results for Algorithm 1 in the following theorem. The results for the gradient dominant situation (P-L condition) is provided in Appendix C. Both of them show that variance reduction can help to improve the convergence of almost all mirror descent algorithms.

Algorithm 1 General Adaptive SMD with Variance Reduction Algorithm

Input: Number of stages T, initial x_1, step sizes $\{\alpha_t\}_{t=1}^T$, batch, mini-batch sizes $\{B_t, b_t\}_{t=1}^T$
for $t = 1$ **to** T **do**
  Randomly sample a batch $\mathcal{I}_t$ with size B_t
  $g_t = \nabla f_{\mathcal{I}_t}(x_t)$; $y_1^t = x_t$
  for $k = 1$ **to** K **do**
    Randomly pick sample $\tilde{\mathcal{I}}_t$ of size b_t
    $v_k^t = \nabla f_{\tilde{\mathcal{I}}_t}(y_k^t) - \nabla f_{\tilde{\mathcal{I}}_t}(y_1^t) + g_t$
    $y_{k+1}^t = \text{argmin}_y\{\alpha_t\langle v_k^t, y\rangle + \alpha_t h(x) + B_{\psi_{tk}}(y, y_k^t)\}$
  end for
  $x_{t+1} = y_{K+1}^t$
end for
Return (Smooth case) Uniformly sample t^* from $\{t\}_{t=1}^T$ and output x_{t^*}; (P-L case) $x_{t^*} = x_{T+1}$

Theorem 1 *Suppose that f satisfies the Lipschitz gradients and bounded variance assumptions **A2, A3** and $\psi_{tk}(x)$ satisfy the m-strong convexity assumption **A1**. Further assume that the learning rate, the batch sizes, the mini-batch sizes, the number of outer and inner loop iterations are set to be $\alpha_t = m/L, B_t = n \wedge (20\sigma^2/m^2\epsilon^2), b_t = b, T = 1 \vee 16\Delta_F L/(m^2\epsilon^2 K), K = \left\lfloor \sqrt{b/20} \right\rfloor \vee 1$, where Δ_F is a constant. Then the output of algorithm 1 converges with gradient computations*

$$O(\frac{n}{\epsilon^2\sqrt{b}} \wedge \frac{\sigma^2}{\epsilon^4\sqrt{b}} + \frac{b}{\epsilon^2})$$

Remark. The proof is relegated to Appendix B. The theorem essentially states with assumption **A1, A2**, and **A3**, we can guarantee the convergence of Algorithm 4. Moreover, similar to ProxSVRG+, when SCSG [17] and ProxSVRG [22] achieve their best convergence at $b = 1$ and $b = n^{2/3}$, our algorithm achieve the best results using a moderate mini-batch size, as shown in corollary 2.

Although α_t is fixed in the theorem, ψ_{tk} can change with time and hence results in the adaptivity. To name an example, using different designs of time-varying step sizes in ProxSVRG+ is allowed (similar to the third example in section 2). When we take the proximal function to be $\psi_{tk}(x) = \frac{c_{tk}}{2}\|x\|_2^2, c_{tk} \geq m$, Algorithm 1 reduces to ProxSVRG+ with time-varying effective step size α_t/c_{tk} (i.e. η in Li and Li [18]). As long as the effective step sizes α_t/c_{tk} are upper bounded by $(m/L)/m = 1/L$, Algorithm 1 still convergences with the same complexity. The upper bound condition is easy to satisfy when using decreasing step sizes, cyclic step sizes [19] or warm up [11]. Besides, ψ_{tk} can be more complicated, such as $\psi_{tk}(x) = \phi_{tk}(x) + \frac{c}{2}\|x\|_2^2, c > 0$, where each $\phi_{tk}(x)$ is an arbitrary convex function, or $\psi_{tk}(x) = \frac{1}{2}\langle x, H_{tk}x\rangle$ as in adaptive algorithms.

Another interesting result observed in our theorem is that when m is small, we require relatively larger batch sizes B_t to guarantee the fast convergence. We show this intuition is actually supported by our experiments in section 4. Next, we show that the convergence can be made faster than the original SMD algorithm by tuning the mini-batch sizes b. We provide the following corollary.

Corollary 2 *With all the assumptions and parameter settings in Theorem 1, further assume that $b = \epsilon^{-4/3}$, where $\epsilon^{-4/3} \leq n$. Then the output of algorithm 1 converges with gradient computations*

$$O(\frac{n}{\epsilon^{4/3}} \wedge \frac{1}{\epsilon^{10/3}} + \frac{1}{\epsilon^{10/3}}) \tag{3}$$

Remark. The above gradient complexity is the same as the best convergence result of ProxSVRG+, and it is provably better than the complexity in equation (2). Therefore, we conclude that variance reduction can indeed reduce the complexity of any adaptive SMD algorithm.

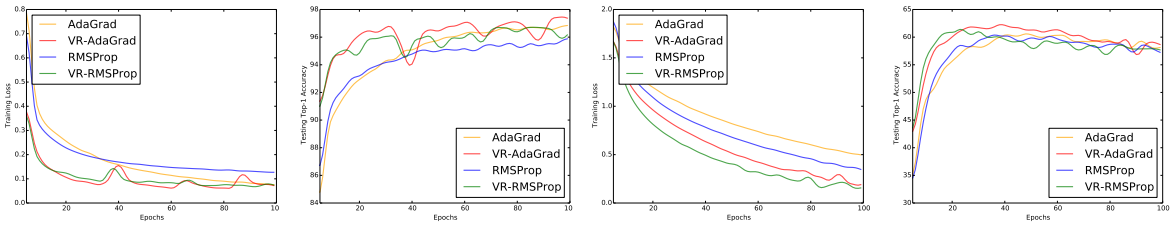
3.2. Extension to Adaptive Subgradient Algorithms

As we have mentioned in section 2, adaptive algorithms such as AdaGrad are special cases of the general adaptive SMD algorithms. The proximal function of adaptive methods is $\psi_t(x) = \frac{1}{2}\langle x, H_t x\rangle$, where H_t is often a diagonal matrix. Assumption **A1** is satisfied because we consistently add a constant m to the matrix H_t. Therefore the conclusions in Theorem 1 still hold for adaptive algorithms. We provide the implementation for Variance Reduced AdaGrad (VR-AdaGrad) in algorithm 4 in Appendix D and Variance Reduced RMSProp (VR-RMSProp) is similar.

However, notice that the strong convexity of these algorithms is relatively weak (m is often set as 1e-3 or even smaller in real experiments), Theorem 1 implies that the batch size B_t needs to be sufficiently large for these algorithm to converge. If variance reduction can work with such algorithms with small m, then we should expect good performances with the other algorithms.

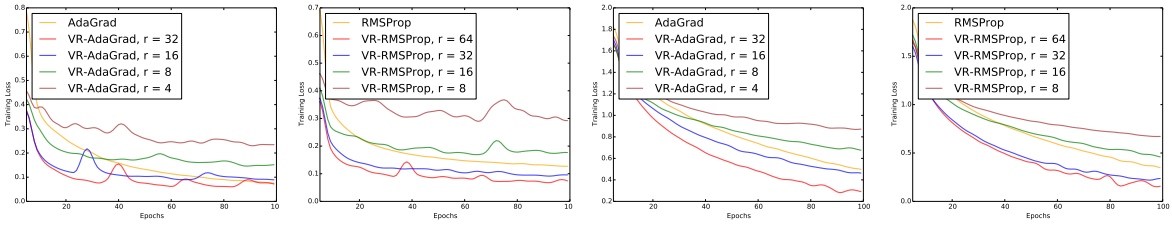
4. Experiments

In this section, we present several experiments on neural networks to show the effectiveness of variance reduction in adaptive SMD algorithms. We choose VR-AdaGrad and VR-RMSProp as two examples of our general algorithm because they have relatively smaller lower bound of the strong convexity. We train a fully connected network on the MNIST dataset and the LeNet [16] on the



(a) MNIST Training Loss (b) MNIST Testing Acc. (c) CIFAR-10 Training Loss. (d) CIFAR-10 Testing Acc.

Figure 1: **(a) and (b)**: training loss and testing accuracy using fully connected network on MNIST. **(c) and (d)**: training loss and testing accuracy using LeNet on CIFAR-10. The results were averaged over five runs.



(a) AdaGrad on MNIST (b) RMSProp on MNIST (c) AdaGrad on CIFAR-10 (d) RMSProp on CIFAR-10

Figure 2: **(a) and (b)**: training loss with different r on MNIST using AdaGrad and RMSProp. **(c) and (d)**: training loss with different r on CIFAR-10 using AdaGrad and RMSProp.

CIFAR-10 dataset. Our implementation is based on the publicly available PyTorch code by yueqiw [25]. For the batch sizes and mini batch sizes B_t and b_t in VR-AdaGrad, VR-RMSProp, we used a slightly different notation of batch size ratio $r = B_t/b_t$. More details of parameter tuning and the neural network model can be found in Appendix D. All the results in Figure 1 and Figure 2 are averaged over five indepedent runs

As can be observed in Figure 1, the variance reduced algorithms converged faster than their original algorithms and their best testing top-1 accuracy was also higher, proving the effectiveness of variance reduction. Some other experiments of using different step sizes are provided in Appendix D. We emphasize that the experiments are not designed to pursue the state-of-the-art performances, but to show that variance reduction can work well with any adaptive proximal functions and lead to faster training, even if the algorithms has very weak convexity guarantees.

Next, we show that algorithms with weaker convexity need a larger batch size B to converge fast. We fixed the mini batch sizes b_t to be the same as in Figure 1 and gradually decreased the batch size ratio r. The baseline ratios of ProxSVRG+ were provided in Appendix D and the performances of VR-AdaGrad and VR-RMSProp were shown in Figure 2. Note that ProxSVRG+ only needed a small ratio ($r = 4$) to be faster than SGD [18], but for VR-AdaGrad and VR-RMSProp, even when $r = 16$, the algorithms still did not converge faster than their original algorithms.

References

[1] Zeyuan Allen-Zhu. Natasha: Faster non-convex stochastic optimization via strongly non-convex parameter, 2017a.

[2] Zeyuan Allen-Zhu. Natasha 2: Faster non-convex optimization than sgd, 2017b.

[3] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 2003.

[4] Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithm for non-convex optimization. *Proceedings of 7th International Conference on Learning Representations(ICLR)*, 2019.

[5] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in Neural Information Processing Systems 27*, 2014.

[6] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. *In Proceedings of the Twenty Third Annual Conference on Computational Learning Theory*, 2010.

[7] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research (JMLR)*, pages 12:2121–2159, 2011.

[8] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems 31*, 2018.

[9] Rong Ge, Zhize Li, Weiyao Wang, and Xiang Wang. Stabilized svrg: Simple variance reduction for nonconvex optimization, 2019.

[10] Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *arXiv preprint arXiv:1308.6594*, 2016.

[11] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2017.

[12] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in Neural Information Processing Systems 27*, 2013.

[13] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-Łojasiewicz condition, 2016.

[14] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[15] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *1*, 2009.

[16] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[17] Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via scsg methods. *Advances in Neural Information Processing Systems 30*, pages 2348–2358, 2017.

[18] Zhize Li and Jian Li. A simple proximal stochastic gradient method for nonsmooth nonconvex optimization. *Advances in Neural Information Processing Systems 31*, pages 5564–5574, 2018.

[19] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *International Conference on Learning Representations*, 2016.

[20] Vinod Nair and Geoffrey Hinton. Rectified linear units improve restricted boltzmann machines. *Proceedings of 27th International Conference on Machine Learning(ICML)*, 2010.

[21] Boris Teodorovich Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 1963.

[22] Sashank Reddi, Suvrit Sra, Barnabas Poczos, and Alexander J Smola. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. *Advances in Neural Information Processing Systems 29*, 2016b.

[23] Bernhard Schölkopf and Alexander J Smola. Learning with kernels: support vector machines, regularization, optimization, and beyond. *MIT press*, 2002.

[24] Tijmen Tieleman and Geoffrey Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, pages 4(2):26–31, 2012.

[25] yueqiw. Svrg for neural networks (pytorch), 2019. URL https://github.com/yueqiw/OptML-SVRG-PyTorch.

[26] Dongruo Zhou, Yiqi Tang, Ziyan Yang, Yuan Cao, and Quanquan Gu. On the convergence of adaptive gradient methods for nonconvex optimization, 2018.

[27] Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic nested variance reduction for nonconvex optimization. *Advances in Neural Information Processing Systems 32*, 2018.

Appendix A. Convergence of Mini-batch Adaptive Mirror Descent

Further notations: We denote the global minimum of $F(x)$ to be $F(x^*)$, and define $\Delta_F = F(x_1) - F(x^*)$, where x_1 is the initialization point of the algorithm. We define the generalized stochastic gradient at t as

$$\tilde{g}_{X,t} = \frac{1}{\alpha_t}(x_t - x_{t+1})$$

Table 1: Comparisons of SFO complexity of different algorithms to reach ϵ-stationary point of the generalized gradient. n is the total number of samples and b is the mini-batch size. "VR" in the last row stands for Variance Reduction. $\tilde{O}$ notation omits the logarithm term $\log \frac{1}{\epsilon}$

Algorithms	Nonconvex Nonsmooth	PL condition
ProxGD [10]	$O(\frac{n}{\epsilon^2})$	$\tilde{O}(\frac{n}{\mu})$
ProxSVRG [22]	$O(\frac{n}{\epsilon^2\sqrt{b}} + n)$	$\tilde{O}(\frac{n}{\mu\sqrt{b}} + n)$
SCSG [17]	$O(\frac{n^{2/3}b^{1/3}}{\epsilon^2} \wedge \frac{b^{1/3}}{\epsilon^{10/3}})$	$\tilde{O}(\frac{nb^{1/3}}{\mu} \wedge \frac{b^{1/3}}{\mu^{5/3}\epsilon^{2/3}} + n \wedge \frac{1}{\mu\epsilon})$
ProxSVRG+ [18]	$O(\frac{n}{\epsilon^2\sqrt{b}} \wedge \frac{1}{\epsilon^4\sqrt{b}} + \frac{b}{\epsilon^2})$	$\tilde{O}((n \wedge \frac{1}{\mu\epsilon})\frac{1}{\mu\sqrt{b}} + \frac{b}{\mu})$
Adaptive SMD (Algorithm 2)	$O(\frac{n}{\epsilon^2} \wedge \frac{1}{\epsilon^4})$	$\tilde{O}(\frac{n}{\mu} \wedge \frac{1}{\mu^2\epsilon})$
Adaptive SMD + VR (Algorithm 1)	$O(\frac{n}{\epsilon^2\sqrt{b}} \wedge \frac{1}{\epsilon^4\sqrt{b}} + \frac{b}{\epsilon^2})$	$\tilde{O}((n \wedge \frac{1}{\mu\epsilon})\frac{1}{\mu\sqrt{b}} + \frac{b}{\mu})$

We first present the convergence rate of Algorithm 2 in the non-convex setting. Despite a few recent analyses on the convergence of SMD and adaptive algorithms such as Ghadimi et al. [10], Zhou et al. [26] and Chen et al. [4], the results on general adaptive SMD is still somewhat lacking. Here we provide the convergence rate of algorithm 2 in Theorem 3.

Algorithm 2 General Adaptive SMD Algorithm

1: **Input:** Number of stages T, initial x_1, step sizes $\{\alpha_t\}_{t=1}^T$
2: **for** $t = 1$ **to** T **do**
3: Randomly sample a batch $\mathcal{I}_t$ with size b
4: $g_t = \nabla f_{\mathcal{I}_t}(x_t)$
5: $x_{t+1} = \text{argmin}_x\{\alpha_t\langle g_t, x\rangle + \alpha_t h(x) + B_{\psi_t}(x, x_t)\}$
6: **end for**
7: **Return** Uniformly sample t^* from $\{t\}_{t=1}^T$ and ouput x_{t^*}

Theorem 3 *Suppose that f satisfies the Lipschitz gradients and bounded variance assumptions **A2, A3**, and $\psi_t(x)$ satisfy the m-strong convexity assumption **A1**. Further assume that the learning rate, the mini batch sizes, and the number of iterations are set to be $\alpha_t = m/L, b = n \wedge (12\sigma^2/(m^2\epsilon^2)), T = 1 \vee (8\Delta_F L/(m^2\epsilon^2))$. Then the output of algorithm 2 converges with gradient computations*

$$O(\frac{n}{\epsilon^2} \wedge \frac{\sigma^2}{\epsilon^4} + n \wedge \frac{\sigma^2}{\epsilon^2}) \tag{4}$$

Remark. Note that if we treat σ^2 as a constant, then the above complexity can be treated as $O(n\epsilon^{-2} \wedge \epsilon^{-4})$. Similar to the results proved by Ghadimi et al. [10], Algorithm 2 needs a relatively large batch size ($O(\epsilon^{-2})$) to obtain a convergence rate close to that of GD ($O(n\epsilon^{-2})$) and SGD ($O(\epsilon^{-4})$). The major reason why algorithm 2 has an advantage over GD and SGD is that we use batched gradient instead of full gradient or stochastic gradient in line 4. However, it is still only asymptotically as fast as one of them, depending on the sample size n.

A.1. Auxiliary Lemmas for Theorem 3

Lemma 4 ***[Lemma 1 in Ghadimi et al. [10]]***. *Let g_t be the stochastic gradient in algorithm 2 obtained at t and $\tilde{g}_{X,t}$ be defined as in (4), then*

$$\langle g_t, \tilde{g}_{X,t} \rangle \geq m\|\tilde{g}_{X,t}\|^2 + \frac{1}{\alpha_t}[h(x_{t+1}) - h(x)] \tag{5}$$

Proof. By the optimality of the mirror descent update rule, it implies for any $x \in \mathcal{X}$ and $\nabla h(x_{t+1}) \in \partial h(x_{t+1})$

$$\langle g_t + \frac{1}{\alpha_t}(\nabla\psi_t(x_{t+1}) - \nabla\psi_t(x_t)) + \nabla h(x_{t+1}), x - x_{t+1} \rangle \geq 0 \tag{6}$$

Let $x = x_t$ in the above in equality, we get

$$\begin{aligned} \langle g_t, x_t - x_{t+1} \rangle &\geq \frac{1}{\alpha_t}\langle \nabla\psi_t(x_{t+1}) - \nabla\psi_t(x_t), x_{t+1} - x_t \rangle + \langle \nabla h(x_{t+1}), x_{t+1} - x_t \rangle \\ &\geq \frac{m}{\alpha_t}\|x_{t+1} - x_t\|_2^2 + h(x_{t+1}) - h(x) \end{aligned} \tag{7}$$

where the second inequality is due to the strong convexity of the function $\psi_t(x)$ and the convexity of $h(x)$, by noting that $x_t - x_{t+1} = \alpha_t \tilde{g}_{X,t}$, the inequality follows.

Lemma 5 *Let $g_{X,t}, \tilde{g}_{X,t}$ be defined as in (4) and (2) respectively, then*

$$\|g_{X,t} - \tilde{g}_{X,t}\|_2 \leq \frac{1}{m}\|\nabla f(x_t) - g_t\|_2 \tag{8}$$

Proof. By definition of g_{X_t} and $\tilde{g}_{X,t}$,

$$\|g_{X,t} - \tilde{g}_{X,t}\|_2 = \frac{1}{\alpha_t}\|(x_t - x_{t+1}^+) - (x_t - x_{t+1})\|_2 = \frac{1}{\alpha_t}\|x_{t+1} - x_{t+1}^+\|_2 \tag{9}$$

Similar to Lemma 4, by the optimality of the mirror descent update rule, we have the following two inequalities

$$\begin{aligned} &\langle g_t + \frac{1}{\alpha_t}(\nabla\psi_t(x_{t+1}) - \nabla\psi_t(x_t)) + \nabla h(x_{t+1}), x - x_{t+1} \rangle \geq 0, \forall x \in \mathcal{X}, \nabla h(x_{t+1}) \in \partial h(x_{t+1}) \\ &\langle \nabla f(x_t) + \frac{1}{\alpha_t}(\nabla\psi_t(x_{t+1}^+) - \nabla\psi_t(x_t)) + \nabla h(x_{t+1}^+), x - x_{t+1}^+ \rangle \geq 0, \forall x \in \mathcal{X}, \nabla h(x_{t+1}^+) \in \partial h(x_{t+1}^+) \end{aligned} \tag{10}$$

Take $x = x_{t+1}^+$ in the first inequality and $x = x_{t+1}$ in the second one, we can get

$$\begin{aligned}\langle g_t, x_{t+1} - x_{t+1}^+\rangle &\geq \frac{1}{\alpha_t}\langle \nabla\psi_t(x_{t+1}) - \nabla\psi_t(x_t), x_{t+1} - x_{t+1}^+\rangle + h(x_{t+1}) - h(x_{t+1}^+)\\ \langle \nabla f(x_t), x_{t+1} - x_{t+1}^+\rangle &\geq \frac{1}{\alpha_t}\langle \nabla\psi_t(x_{t+1}^+) - \nabla\psi_t(x_t), x_{t+1}^+ - x_{t+1}\rangle + h(x_{t+1}^+) - h(x_{t+1})\end{aligned} \tag{11}$$

Summing up the above inequalities, we can get

$$\begin{aligned}&\langle g_t - \nabla f(x_t), x_{t+1} - x_{t+1}^+\rangle\\ &\geq \frac{1}{\alpha_t}(\langle \nabla\psi_t(x_{t+1}) - \nabla\psi_t(x_t), x_{t+1} - x_{t+1}^+\rangle + \langle \nabla\psi_t(x_{t+1}^+) - \nabla\psi_t(x_t), x_{t+1}^+ - x_{t+1}\rangle)\\ &= \frac{1}{\alpha_t}(\langle \nabla\psi_t(x_{t+1}) - \nabla\psi_t(x_{t+1}^+), x_{t+1} - x_{t+1}^+\rangle)\\ &\geq \frac{m}{\alpha_t}\|x_{t+1} - x_{t+1}^+\|_2^2\end{aligned} \tag{12}$$

Therefore by Cauchy Schwarz inequality,

$$\|g_t - \nabla f(x_t)\|_2 \geq \frac{m}{\alpha_t}\|x_{t+1} - x_{t+1}^+\|_2 \tag{13}$$

Hence the inequality in the lemma follows.

Lemma 6 ***[Lemma A.1 in Lei et al. [17]].*** *Let $x_1, \cdots, x_M \in \mathbb{R}^d$ be an arbitrary population of M vectors with the condition that*

$$\sum_{i=1}^{M} x_j = 0 \tag{14}$$

Further let J be a uniform random subset of $\{1, \cdots, M\}$ with size m, then

$$\mathbb{E}[\|\frac{1}{m}\sum_{j\in J} x_j\|^2] \leq \frac{I(m < M)}{mM}\sum_{j=1}^{M}\|x_j\|^2 \tag{15}$$

Proof of the above general lemma can be found in Lei et al. [17].

A.2. Proof of the Convergence of the Adaptive SMD Algorithm (Theorem 3)

Proof. From the L-Lipshitz gradients and Lemma 4, we know that

$$\begin{aligned}f(x_{t+1}) &\leq f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t\rangle + \frac{L}{2}\|x_{t+1} - x_t\|^2\\ &= f(x_t) - \alpha_t\langle \nabla f(x_t), \tilde{g}_{X,t}\rangle + \frac{L}{2}\alpha_t^2\|\tilde{g}_{X,t}\|_2^2\\ &= f(x_t) - \alpha_t\langle g_t, \tilde{g}_{X,t}\rangle + \frac{L}{2}\alpha_t^2\|\tilde{g}_{X,t}\|_2^2 + \alpha_t\langle g_t - \nabla f(x_t), \tilde{g}_{X,t}\rangle\\ &\leq f(x_t) + \frac{L}{2}\alpha_t^2\|\tilde{g}_{X,t}\|_2^2 - \alpha_t m\|\tilde{g}_{X,t}\|_2^2 - [h(x_{t+1}) - h(x)] + \alpha_t\langle g_t - \nabla f(x_t), \tilde{g}_{X,t}\rangle\end{aligned} \tag{16}$$

Therefore since $F(x) = f(x) + h(x)$, we get

$$\begin{aligned} F(x_{t+1}) &\leq F(x_t) - (\alpha_t m - \frac{L}{2}\alpha_t^2)\|\tilde{g}_{X,t}\|_2^2 + \alpha_t\langle g_t - \nabla f(x_t), \tilde{g}_{X,t}\rangle + \alpha_t\langle g_t - \nabla f(x_t), \tilde{g}_{X,t} - g_{X,t}\rangle \\ &\leq F(x_t) - (\alpha_t m - \frac{L}{2}\alpha_t^2)\|\tilde{g}_{X,t}\|_2^2 + \alpha_t\langle g_t - \nabla f(x_t), \tilde{g}_{X,t}\rangle + \alpha_t\|\nabla f(x_t) - g_t\|_2\|\tilde{g}_{X,t} - g_{X,t}\|_2 \\ &\leq F(x_t) - (\alpha_t m - \frac{L}{2}\alpha_t^2)\|\tilde{g}_{X,t}\|_2^2 + \alpha_t\langle g_t - \nabla f(x_t), \tilde{g}_{X,t}\rangle + \frac{\alpha_t}{m}\|\nabla f(x_t) - g_t\|_2^2 \end{aligned} \tag{17}$$

where the second last one is a direct result from Cauchy-Schwarz inequality and the last inequality is from Lemma 5. Rearrange the above inequalities and sum up from 1 to T, we get

$$\begin{aligned} \sum_{t=1}^{T}(\alpha_t m - \frac{L}{2}\alpha_t^2)\|\tilde{g}_{X,t}\|_2^2 &\leq \sum_{t=1}^{T}[F(x_t) - F(x_{t+1})] + \sum_{t=1}^{T}[\alpha_t\langle g_t - \nabla f(x_t), \tilde{g}_{X,t}\rangle + \frac{\alpha_t}{m}\|\nabla f(x_t) - g_t\|_2^2] \\ &= F(x_1) - F(x_{T+1}) + \sum_{t=1}^{T}[\alpha_t\langle g_t - \nabla f(x_t), \tilde{g}_{X,t}\rangle + \frac{\alpha_t}{m}\|\nabla f(x_t) - g_t\|^2] \\ &\leq F(x_1) - F(x^*) + \sum_{t=1}^{T}[\alpha_t\langle g_t - \nabla f(x_t), \tilde{g}_{X,t}\rangle + \frac{\alpha_t}{m}\|\nabla f(x_t) - g_t\|^2] \end{aligned} \tag{18}$$

where the last inequality is due to $f(x^*) \leq f(x), \forall x$. Define the filtration $\mathcal{F}_t = \sigma(x_1, \cdots, x_t)$. Note that we suppose g_t is an unbiased estimate of $\nabla f(x_t)$, hence $\mathbb{E}[\langle \nabla f(x_t) - g_t, g_{X,t}\rangle | \mathcal{F}_t] = 0$. Moreover, since the sampled gradients has bounded variance σ^2, hence by applying Lemma 6 with $x_i = \nabla_{i\in\mathcal{I}_j} f_i(x_t) - \nabla f(x_t)$

$$\mathbb{E}[\|\nabla f(x_t) - g_t\|^2] \leq \frac{\sigma^2}{b_t} I(b_t < n) \tag{19}$$

where I is the indicator function. Since the final x_{t^*} is uniformly sampled from all $\{x_t\}_{t=1}^T$, therefore

$$\mathbb{E}[\|\tilde{g}_{X,t^*}\|_2^2] = \mathbb{E}[\mathbb{E}[\|\tilde{g}_{X,t^*}\|_2^2 | t^*]] = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\tilde{g}_{X,t}\|_2^2] \tag{20}$$

Therefore when α_t, b_t are constants, the average can be found as

$$\begin{aligned} T(\alpha_t m - \frac{L}{2}\alpha_t^2)\mathbb{E}(\|\tilde{g}_{X,t^*}\|_2^2) &\leq F(x_1) - F(x^*) + \sum_{t=1}^{T}\frac{\alpha_t}{m}\mathbb{E}[\|\nabla f(x_t) - g_t\|_2^2] \\ &= \Delta_F + T\frac{\alpha_t\sigma^2}{m b_t} I(b_t < n) \end{aligned} \tag{21}$$

where we define $\Delta_F = F(x_1) - F(x^*)$. Take $\alpha_t = \frac{m}{L}$, then $\alpha_t m - \frac{L}{2}\alpha_t^2 = \frac{m^2}{2L}$ and

$$\mathbb{E}(\|\tilde{g}_{X,t^*}\|_2^2) \leq \frac{2\Delta_F L}{m^2 T} + \frac{2\sigma^2}{b_t m^2} I(b_t < n) \tag{22}$$

Also by Lemma 5, the difference between g_{X,t^*} and $\tilde{g}_{X,t^*}$ are bounded, hence

$$\begin{aligned}\mathbb{E}[\|g_{X,t^*}\|_2^2] &\leq 2\mathbb{E}[\|\tilde{g}_{X,t^*}\|_2^2] + 2\mathbb{E}[\|g_{X,t^*} - \tilde{g}_{X,t^*}\|_2^2] \\ &\leq \frac{4\Delta_F L}{m^2 T} + \frac{4\sigma^2}{b_t m^2} I(b_t < n) + \frac{2\sigma^2}{b_t m^2} I(b_t < n) \\ &= \frac{4\Delta_F L}{m^2 T} + \frac{6\sigma^2}{b_t m^2} I(b_t < n)\end{aligned} \tag{23}$$

Take $b_t = n \wedge (12\sigma^2/m^2\epsilon^2), T = 1 \vee (8\Delta_F L/m^2\epsilon^2)$ as in the theorem, the expectation is

$$\begin{aligned}\mathbb{E}[\|g_{X,t^*}\|_2^2] &\leq \frac{4\Delta_F L}{m^2 T} + \frac{6\sigma^2}{b_t m^2} I(b_t < n) \\ &\leq \frac{\epsilon^2}{2} + \frac{\epsilon^2}{2} = \epsilon^2\end{aligned} \tag{24}$$

Therefore since one iteration takes b_t stochastic gradient computations, the total number of stochastic gradient computations is

$$T b_t \leq \frac{8\Delta_F L}{m^2\epsilon^2} b_t + b_t = O(\frac{n}{\epsilon^2} \wedge \frac{\sigma^2}{\epsilon^4} + n \wedge \frac{\sigma^2}{\epsilon^2}) \tag{25}$$

A.3. Convergence of Algorithm 2 under the PL condition

By the proof in A.2, we have

$$F(x_{t+1}) \leq F(x_t) - (\alpha_t m - \frac{L}{2}\alpha_t^2)\|\tilde{g}_{X,t}\|_2^2 + \alpha_t\langle g_t - \nabla f(x_t), \tilde{g}_{X,t}\rangle + \frac{\alpha_t}{m}\|\nabla f(x_t) - g_t\|_2^2 \tag{26}$$

Take expectation on both sides, we know that

$$\mathbb{E}[F(x_{t+1})] \leq \mathbb{E}[F(x_t)] - (\alpha_t m - \frac{L}{2}\alpha_t^2)\mathbb{E}[\|\tilde{g}_{X,t}\|_2^2] + \frac{\alpha_t}{m}\mathbb{E}[\|\nabla f(x_t) - g_t\|_2^2] \tag{27}$$

Since

$$\mathbb{E}[\|g_{X,t^*}\|_2^2] \leq 2\mathbb{E}[\|\tilde{g}_{X,t^*}\|_2^2] + 2\mathbb{E}[\|g_{X,t^*} - \tilde{g}_{X,t^*}\|_2^2] \tag{28}$$

Hence the inequality becomes

$$\begin{aligned}\mathbb{E}[F(x_{t+1})] &\leq \mathbb{E}[F(x_t)] - (\alpha_t m - \frac{L}{2}\alpha_t^2)(\frac{1}{2}\mathbb{E}[\|g_{X,t}\|_2^2] - \mathbb{E}[\|\nabla f(x_t) - g_t\|_2^2]) + \frac{\alpha_t}{m}\mathbb{E}[\|\nabla f(x_t) - g_t\|_2^2] \\ &\leq \mathbb{E}[F(x_t)] - (\frac{\alpha_t m}{2} - \frac{L}{4}\alpha_t^2)\mathbb{E}[\|g_{X,t}\|_2^2] + (\frac{\alpha_t}{m} + \alpha_t m - \frac{L}{2}\alpha_t^2)\mathbb{E}[\|\nabla f(x_t) - g_t\|_2^2] \\ &\leq \mathbb{E}[F(x_t)] - \mu(\alpha_t m - \frac{L}{2}\alpha_t^2)(\mathbb{E}[F(x_t)] - F(x^*)) + (\frac{\alpha_t}{m} + \alpha_t m - \frac{L}{2}\alpha_t^2)\mathbb{E}[\|\nabla f(x_t) - g_t\|_2^2]\end{aligned} \tag{29}$$

Take $\alpha_t = m/L$ and minus $F(x)^*$ on both sides, we get

$$\begin{aligned}\mathbb{E}[F(x_{t+1})] - F(x^*) &\leq (1-\mu(\alpha_t m - \frac{L}{2}\alpha_t^2)(\mathbb{E}[F(x_t)] - F(x^*)) + (\frac{\alpha_t}{m} + \alpha_t m - \frac{L}{2}\alpha_t^2)\mathbb{E}[\|\nabla f(x_t) - g_t\|_2^2] \\ &= (1-\mu\frac{m^2}{2L})(\mathbb{E}[F(x_t)] - F(x^*)) + (\frac{1}{L} + \frac{m^2}{2L})\mathbb{E}[\|\nabla f(x_t) - g_t\|_2^2] \\ &= (1-\mu\frac{m^2}{2L})(\mathbb{E}[F(x_t)] - F(x^*)) + (\frac{1}{L} + \frac{m^2}{2L})\frac{\sigma^2}{b_t} I(b_t < n)\end{aligned} \tag{30}$$

Let $\gamma = 1 - \frac{\mu m^2}{2L}$, since $m^2\mu/L \leq \frac{1}{\sqrt{n}}$, $\gamma \in (0,1)$, divide by γ^{t+1} on both sides, we get

$$\frac{\mathbb{E}[F(x_{t+1})] - F(x^*)}{\gamma^{t+1}} \leq \frac{\mathbb{E}[F(x_t)] - F(x^*)}{\gamma^t} + \frac{(\frac{1}{L} + \frac{m^2}{2L})}{\gamma^{t+1}}\frac{\sigma^2}{b_t} I(b_t < n) \tag{31}$$

Take summation with respect to the loop parameter t from $t = 1$ to $t = T$, assume that b_t is a constant, the inequality becomes

$$\begin{aligned}\mathbb{E}[F(x_{T+1})] - F(x^*) &\leq \gamma^T \Delta_F + \gamma^T \sum_{t=1}^{T} \frac{(\frac{1}{L} + \frac{m^2}{2L})}{\gamma^t}\frac{\sigma^2}{b_t} I(b_t < n) \\ &\leq \gamma^T \Delta_F + (\frac{1}{L} + \frac{m^2}{2L})\frac{1-\gamma^T}{1-\gamma}\frac{\sigma^2}{b_t} I(b_t < n) \\ &\leq \gamma^T \Delta_F + (\frac{1}{L} + \frac{m^2}{2L})\frac{2L}{\mu m^2}\frac{\sigma^2}{b_t} I(b_t < n) \\ &= \gamma^T \Delta_F + (\frac{1}{m^2} + 1)\frac{1}{\mu}\frac{\sigma^2}{b_t} I(b_t < n)\end{aligned} \tag{32}$$

Therefore when taking $T = 1 \vee (\log \frac{2\Delta_F}{\epsilon})/(\log \frac{1}{\gamma}) = O(\log \frac{2\Delta_F}{\epsilon}/\mu)$, $b_t = n \wedge \frac{2(1+m^2)\sigma^2}{\epsilon m^2 \mu}$. Then the total number of stochastic gradient computations is

$$\begin{aligned}Tb &= O((n \wedge \frac{\sigma^2}{\mu\epsilon})(\frac{1}{\mu}\log\frac{1}{\epsilon})) \\ &= O((\frac{n}{\mu} \wedge \frac{\sigma^2}{\mu^2\epsilon})\log\frac{1}{\epsilon})\end{aligned} \tag{33}$$

Appendix B. Convergence of Adaptive Mirror Descent with Variance Reduction

Recall the algorithm 1 in the algorithm section, similarly define

$$\tilde{g}_{Y,k}^t = \frac{1}{\alpha_t}(y_k^t - y_{k+1}^t) \tag{34}$$

and its corresponding term when the algorithm uses non-stochastic full batch gradient

$$g_{Y,k}^t = \frac{1}{\alpha_t}(y_k^t - y_{k+1}^{t+}), \text{ when } y_{k+1}^{t+} = \text{argmin}_y\{\alpha_t\langle\nabla f(y_k^t), y\rangle + \alpha_t h(x) + B_{\psi_{tk}}(y, y_k^t)\} \tag{35}$$

Algorithm 3 Adaptive SMD with Variance Reduction Algorithm

Input: Number of stages T, initial x_1, step sizes $\{\alpha_t\}_{t=1}^T$, batch sizes $\{B_t\}_{t=1}^T$, mini-batch sizes $\{b_t\}_{t=1}^T$
for $t = 1$ **to** T **do**
  Randomly sample a batch $\mathcal{I}_t$ with size B_t
  $g_t = \nabla f_{\mathcal{I}_t}(x_t)$
  $y_1^t = x_t$
  for $k = 1$ **to** K **do**
    Randomly pick sample $\tilde{\mathcal{I}}_t$ of size b_t
    $v_k^t = \nabla f_{\tilde{\mathcal{I}}_t}(y_k^t) - \nabla f_{\tilde{\mathcal{I}}_t}(y_1^t) + g_t$
    $y_{k+1}^t = \text{argmin}_y\{\alpha_t\langle v_k^t, y\rangle + \alpha_t h(x) + B_{\psi_{tk}}(y, y_k^t)\}$
  end for
  $x_{t+1} = y_{K+1}^t$
end for
Return (Smooth case) Uniformly sample x_{t^*} from $\{y_k^t\}_{k=1,t=1}^{K,T}$; (P-L case) $x_{t^*} = x_{T+1}$

B.1. Auxiliary Lemmas for Theorem 1

Lemma 7 *Let v_k^t be defined as in algorithm 1 and $\tilde{g}_{Y,k}^t$ be defined as in (34), then*

$$\langle v_k^t, \tilde{g}_{Y,k}^t\rangle \geq m\|\tilde{g}_{Y,k}^t\|^2 + \frac{1}{\alpha_t}[h(y_{k+1}^t) - h(y_k^t)] \tag{36}$$

Proof. The proof of this inequality is similar to that of Lemma 4. By the optimality of the mirror descent update rule, it implies for any $y \in \mathcal{X}, \nabla h(y_{k+1}^t) \in \partial h(y_{k+1}^t)$,

$$\langle v_k^t + \frac{1}{\alpha_t}(\nabla\psi_{tk}(y_{k+1}^t) - \nabla\psi_{tk}(y_k^t)) + \nabla h(y_{k+1}^t), y - y_{k+1}^t\rangle \geq 0 \tag{37}$$

Let $x = y_k^t$ in the above in equality, we get

$$\begin{aligned}\langle v_k^t, y_k^t - y_{k+1}^t\rangle &\geq \frac{1}{\alpha_t}\langle\nabla\psi_{tk}(y_{k+1}^t) - \nabla\psi_{tk}(y_k^t), y_{k+1}^t - y_k^t\rangle + \langle\nabla h(y_{k+1}^t), y_{k+1}^t - y_k^t\rangle \\ &\geq \frac{m}{\alpha_t}\|y_{k+1}^t - y_k^t\|_2^2 + [h(y_{k+1}^t) - h(y_k^t)]\end{aligned} \tag{38}$$

where the second inequality is due to the m-strong convexity of the function $\psi_{tk}(x)$ and the convexity of h. Note from the definition that $y_k^t - y_{k+1}^t = \alpha_t\tilde{g}_{Y,k}^t$, the inequality follows.

Lemma 8 *Let $g_{Y,k}^t, \tilde{g}_{Y,k}^t$ be defined as in (34) and (35) respectively, then*

$$\|\tilde{g}_{Y,k}^t - g_{Y,k}^t\|_2 \leq \frac{1}{m}\|\nabla f(y_k^t) - v_k^t\|_2 \tag{39}$$

Proof. The proof is similar to Lemma 5. By definition of $\tilde{g}_{Y,k}^t$ and $g_{Y,k}^t$,

$$\|\tilde{g}_{Y,k}^t - g_{Y,k}^t\|_2 = \frac{1}{\alpha_t}\|(y_k^t - y_{k+1}^t) - (y_k^t - y_{k+1}^{t+})\|_2 = \frac{1}{\alpha_t}\|y_k^t - y_{k+1}^{t+}\|_2 \tag{40}$$

As in Lemma 7, by the optimality of the mirror descent update rule, we have the following two inequalities

$$\begin{aligned}&\langle v_k^t + \frac{1}{\alpha_t}(\nabla\psi_{tk}(y_{k+1}^t) - \nabla\psi_{tk}(y_k^t)) + \nabla h(y_{k+1}^t), y - y_{k+1}^t\rangle \geq 0, \forall y \in \mathcal{X}, \nabla h(y_{k+1}^t) \in \partial h(y_{k+1}^t)\\ &\langle \nabla f(y_k^t) + \frac{1}{\alpha_t}(\nabla\psi_{tk}(y_{k+1}^{t+}) - \nabla\psi_{tk}(y_k^t)) + \nabla h(y_{k+1}^{t+}), y - y_{k+1}^{t+}\rangle \geq 0, \forall y \in \mathcal{X}, \nabla h(y_{k+1}^{t+1}) \in \partial h(y_{k+1}^{t+})\end{aligned} \tag{41}$$

Take $y = y_{k+1}^{t+}$ in the first inequality and $y = y_{k+1}^t$ in the second one, we can get

$$\begin{aligned}&\langle v_k^t, y_{k+1}^{t+} - y_{k+1}^t\rangle \geq \frac{1}{\alpha_t}\langle\nabla\psi_{tk}(y_{k+1}^t) - \nabla\psi_{tk}(y_k^t), y_{k+1}^t - y_{k+1}^{t+}\rangle + h(y_{k+1}^t) - h(y_{k+1}^{t+})\\ &\langle \nabla f(y_k^t), y_{k+1}^t - y_{k+1}^{t+}\rangle \geq \frac{1}{\alpha_t}\langle\nabla\psi_{tk}(y_k^t) - \nabla\psi_{tk}(y_{k+1}^{t+}), y_{k+1}^{t+} - y_{k+1}^t\rangle + h(y_{k+1}^{t+}) - h(y_{k+1}^t)\end{aligned} \tag{42}$$

Summing up the above inequalities, we can get

$$\begin{aligned}&\langle v_k^t - \nabla f(y_k^t), y_{k+1}^{t+} - y_{k+1}^t\rangle\\ &\geq \frac{1}{\alpha_t}\langle\nabla\psi_{tk}(y_{k+1}^t) - \nabla\psi_{tk}(y_k^t), y_{k+1}^t - y_{k+1}^{t+}\rangle + \frac{1}{\alpha_t}\langle\nabla\psi_{tk}(y_k^t) - \nabla\psi_{tk}(y_{k+1}^{t+}), y_{k+1}^{t+} - y_{k+1}^t\rangle\\ &= \frac{1}{\alpha_t}(\langle\nabla\psi_{tk}(y_{k+1}^t) - \nabla\psi_{tk}(y_{k+1}^{t+}), y_{k+1}^t - y_{k+1}^{t+}\rangle)\\ &\geq \frac{m}{\alpha_t}\|y_{k+1}^t - y_{k+1}^{t+}\|_2^2\end{aligned} \tag{43}$$

where the last inequality is due to the strong convexity of $\psi_{tk}(x)$. Therefore by Cauchy Schwarz inequality,

$$\frac{1}{m}\|\nabla f(y_k^t) - v_k^t\|_2 \geq \frac{1}{\alpha_t}\|y_{k+1}^t - y_{k+1}^{t+}\|_2 \geq \|\tilde{g}_{Y,k}^t - g_{Y,k}^t\|_2 \tag{44}$$

Hence the inequality in the lemma follows.

Lemma 9 *Let $\nabla f(y_k^t), v_k^t$ be the full batch gradient and the , then*

$$\mathbb{E}[\|\nabla f(y_k^t) - v_k^t\|_2^2] \leq \frac{L^2}{b_t}\mathbb{E}[\|y_k^t - x_t\|^2] + \frac{I(B_t < n)\sigma^2}{B_t} \tag{45}$$

Proof. Note that the large batch $\mathcal{I}_j$ and the mini-batch $\tilde{\mathcal{I}}_j$ are independent, hence

$$
\begin{aligned}
&\mathbb{E}[\|\nabla f(y_k^t) - v_k^t\|_2^2] \\
&= \mathbb{E}[\|\frac{1}{b_t}\sum_{i\in\tilde{\mathcal{I}}_k}(\nabla f_i(y_k^t) - \nabla f_i(x_t)) - (\nabla f(y_k^t) - g_t)\|_2^2] \\
&= \mathbb{E}[\|\frac{1}{b_t}\sum_{i\in\tilde{\mathcal{I}}_k}(\nabla f_i(y_k^t) - \nabla f_i(x_t)) - (\nabla f(y_k^t) - \frac{1}{B_t}\sum_{i\in\mathcal{I}_t}\nabla f_i(x_t))\|_2^2] \\
&= \mathbb{E}[\|\frac{1}{b_t}\sum_{i\in\tilde{\mathcal{I}}_k}(\nabla f_i(y_k^t) - \nabla f_i(x_t)) - \nabla f(y_k^t) + \nabla f(x_t) + \frac{1}{B_t}\sum_{i\in\mathcal{I}_t}(\nabla f_i(x_t) - \nabla f(x_t))\|_2^2] \\
&= \mathbb{E}[\|\frac{1}{b_t}\sum_{i\in\tilde{\mathcal{I}}_k}(\nabla f_i(y_k^t) - \nabla f_i(x_t)) - \nabla f(y_k^t) + \nabla f(x_t)\|_2^2 + \mathbb{E}\|\frac{1}{B_t}\sum_{i\in\mathcal{I}_t}(\nabla f_i(x_t) - \nabla f(x_t))\|_2^2] \\
&= \mathbb{E}[\|\frac{1}{b_t}\sum_{i\in\tilde{\mathcal{I}}_k}(\nabla f_i(y_k^t) - \nabla f(y_k^t)) - (\nabla f_i(x_t) - \nabla f(x_t))\|_2^2 + \mathbb{E}\|\frac{1}{B_t}\sum_{i\in\mathcal{I}_t}(\nabla f_i(x_t) - \nabla f(x_t))\|_2^2] \\
&\leq \mathbb{E}[\|\frac{1}{b_t}\sum_{i\in\tilde{\mathcal{I}}_k}(\nabla f_i(y_k^t) - \nabla f(y_k^t)) - (\nabla f_i(x_t) - \nabla f(x_t))\|_2^2 + \frac{I(B_t < n)\sigma^2}{B_t} \\
&= \frac{1}{b_t^2}\mathbb{E}[\sum_{i\in\tilde{\mathcal{I}}_k}\|\nabla f_i(y_k^t) - \nabla f_i(x_t)) - \nabla f(y_k^t) + \nabla f(x_t)\|^2] + \frac{I(B_t < n)\sigma^2}{B_t} \\
&\leq \frac{1}{b_t^2}\mathbb{E}[\sum_{i\in\tilde{\mathcal{I}}_k}\|\nabla f_i(y_k^t) - \nabla f_i(x_t))\|^2] + \frac{I(B_t < n)\sigma^2}{B_t} \\
&\leq \frac{L^2}{b_t}\mathbb{E}[\|y_k^t - x_t\|^2] + \frac{I(B_t < n)\sigma^2}{B_t}
\end{aligned}
\tag{46}
$$

where the fourth equality is because of the independence between $\mathcal{I}_j$ and $\tilde{\mathcal{I}}_j$. The first and the second inequalities are by Lemma 6. The third inequality follows from $\mathbb{E}[\|x - \mathbb{E}(x)\|^2] = \mathbb{E}[\|x\|^2]$ and the last inequality follows from the L-smoothness of $f(x)$

B.2. Proof of Convergence of the adaptive SMD with Variance Reduction Algorithm (Theorem 1)

From the L-Lipshitz gradients and Lemma 7, we know that

$$
\begin{aligned}
f(y_{k+1}^t) &\leq f(y_k^t) + \langle\nabla f(y_k^t), y_{k+1}^t - y_k^t\rangle + \frac{L}{2}\|y_{k+1}^t - y_k^t\|^2 \\
&= f(y_k^t) - \alpha_t\langle\nabla f(y_k^t), \tilde{g}_{Y,k}^t\rangle + \frac{L}{2}\alpha_t^2\|\tilde{g}_{Y,k}^t\|_2^2 \\
&= f(y_k^t) - \alpha_t\langle v_k^t, \tilde{g}_{Y,k}^t\rangle + \frac{L}{2}\alpha_t^2\|\tilde{g}_{Y,k}^t\|_2^2 + \alpha_t\langle v_k^t - \nabla f(y_k^t), \tilde{g}_{Y,k}^t\rangle \\
&\leq f(y_k^t) + \frac{L}{2}\alpha_t^2\|\tilde{g}_{Y,k}^t\|_2^2 - \alpha_t m\|\tilde{g}_{Y,k}^t\|_2^2 + \alpha_t\langle v_k^t - \nabla f(y_k^t), \tilde{g}_{Y,k}^t\rangle - [h(y_{k+1}^t) - h(y_k^t)]
\end{aligned}
\tag{47}
$$

Since $F(x) = f(x) + h(x)$, we can get

$$\begin{aligned}
F(y_{k+1}^t) &= F(y_k^t) - (\alpha_t m - \frac{L}{2}\alpha_t^2)\|\tilde{g}_{Y,k}^t\|_2^2 + \alpha_t\langle v_k^t - \nabla f(y_k^t), g_{Y,k}^t\rangle + \alpha_t\langle v_k^t - \nabla f(y_k^t), \tilde{g}_{Y,k}^t - g_{Y,k}^t\rangle \\
&\leq F(y_k^t) - (\alpha_t m - \frac{L}{2}\alpha_t^2)\|\tilde{g}_{Y,k}^t\|_2^2 + \alpha_t\langle v_k^t - \nabla f(y_k^t), g_{Y,k}^t\rangle + \alpha_t\|\nabla f(y_k^t) - v_k^t\|_2\|\tilde{g}_{Y,k}^t - g_{Y,k}^t\|_2 \\
&\leq F(y_k^t) - (\alpha_t m - \frac{L}{2}\alpha_t^2)\|\tilde{g}_{Y,k}^t\|_2^2 + \alpha_t\langle v_k^t - \nabla f(y_k^t), g_{Y,k}^t\rangle + \frac{\alpha_t}{m}\|\nabla f(y_k^t) - v_k^t\|_2^2
\end{aligned} \tag{48}$$

where the second last inequality is from Cauchy Schwartz inequality and the last inequality is from Lemma 8. Define the filtration $\mathcal{F}_k^t = \sigma(y_1^1, \cdots y_{K+1}^1, y_1^2, \cdots, y_{K+1}^2, \cdots, y_1^t, \cdots, y_k^t)$. Note that $\mathbb{E}[\langle \nabla f(y_k^t) - v_k^t, g_{Y,k}^t\rangle | F_k^t] = 0$. Take expectation on both sides and use Lemma 9, we get

$$\begin{aligned}
\mathbb{E}[F(y_{k+1}^t)] &\leq \mathbb{E}[F(y_k^t)] - (\frac{m}{\alpha_t} - \frac{L}{2})\mathbb{E}[\|y_{k+1}^t - y_k^t\|_2^2] + \frac{L^2\alpha_t}{b_t m}\mathbb{E}[\|y_k^t - x_t\|^2] + \frac{\alpha_t I(B_t < n)\sigma^2}{mB_t} \\
&\leq \mathbb{E}[F(y_k^t)] - (\frac{m}{2\alpha_t} - \frac{L}{4})\mathbb{E}[\|y_{k+1}^t - y_k^t\|_2^2] + \frac{L^2\alpha_t}{b_t m}\mathbb{E}[\|y_k^t - x_t\|^2] + \frac{\alpha_t I(B_t < n)\sigma^2}{mB_t} \\
&\quad + (\frac{m}{2\alpha_t} - \frac{L}{4})\frac{\alpha_t^2 L^2}{m^2 b_t}\mathbb{E}[\|y_k^t - x_t\|^2] + (\frac{m}{2\alpha_t} - \frac{L}{4})\frac{\alpha_t^2 I(B_t < n)\sigma^2}{m^2 B_t} - (\frac{m}{4\alpha_t} - \frac{L}{8})\mathbb{E}[\|y_{k+1}^{t+} - y_k^t\|_2^2] \\
&= \mathbb{E}[F(y_k^t)] - (\frac{m}{2\alpha_t} - \frac{L}{4})\mathbb{E}[\|y_{k+1}^t - y_k^t\|_2^2] - (\frac{m}{4\alpha_t} - \frac{L}{8})\mathbb{E}[\|y_{k+1}^{t+} - y_k^t\|_2^2] \\
&\quad + (\frac{3L^2\alpha_t}{2b_t m} - \frac{\alpha_t^2 L^3}{4m^2 b_t})\mathbb{E}[\|y_k^t - x_t\|^2] + (\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t} \\
&= \mathbb{E}[F(y_k^t)] - (\frac{m}{2\alpha_t} - \frac{L}{4})\mathbb{E}[\|y_{k+1}^t - y_k^t\|_2^2] - (\frac{m\alpha_t}{4} - \frac{L\alpha_t^2}{8})\mathbb{E}[\|g_{Y,k}^t\|_2^2] \\
&\quad + (\frac{3L^2\alpha_t}{2b_t m} - \frac{\alpha_t^2 L^3}{4m^2 b_t})\mathbb{E}[\|y_k^t - x_t\|^2] + (\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t}
\end{aligned} \tag{49}$$

where the second inequality uses the fact that by Lemma 9

$$\begin{aligned}
\mathbb{E}[\|y_{k+1}^{t+} - y_k^t\|_2^2] &= \alpha_t^2\mathbb{E}[\|g_{Y,k}^t\|_2^2] \leq 2\alpha_t^2\mathbb{E}[\|\tilde{g}_{Y,k}^t\|_2^2] + 2\alpha_t^2\mathbb{E}[\|g_{Y,k}^t - \tilde{g}_{Y,k}^t\|_2^2] \\
&\leq 2\alpha_t^2\mathbb{E}[\|\tilde{g}_{Y,k}^t\|_2^2] + \frac{2\alpha_t^2}{m^2}\mathbb{E}[\|\nabla f(y_k^t) - v_k^t\|_2^2] \\
&\leq 2\mathbb{E}[\|y_{k+1}^t - y_k^t\|_2^2] + \frac{2\alpha_t^2}{m^2}(\frac{L^2}{b_t}\mathbb{E}[\|y_k^t - x_t\|^2] + \frac{I(B_t < n)\sigma^2}{B_t}) \\
&= 2\mathbb{E}[\|y_{k+1}^t - y_k^t\|_2^2] + \frac{2\alpha_t^2 L^2}{m^2 b_t}\mathbb{E}[\|y_k^t - x_t\|^2] + \frac{2I(B_t < n)\sigma^2\alpha_t^2}{B_t m^2}
\end{aligned} \tag{50}$$

Since by Young's inequality, we know that

$$\|y_{k+1}^t - x_t\| \leq (1 + \frac{1}{p})\|y_k^t - x_t\|_2^2 + (1+p)\|y_{k+1}^t - y_k^t\|_2^2, \forall p \in \mathbb{R} \tag{51}$$

Hence substitute into equation 49, we can get

$$\begin{aligned}\mathbb{E}[F(y_{k+1}^t)] &\leq \mathbb{E}[F(y_k^t)] - (\frac{m}{2\alpha_t} - \frac{L}{4})\mathbb{E}(\frac{\|y_{k+1}^t - x_t\|_2^2}{1+p} - \frac{\|y_k^t - x_t\|_2^2}{p}) - (\frac{m\alpha_t}{4} - \frac{L\alpha_t^2}{8})\mathbb{E}[\|g_{Y,k}^t\|_2^2] \\ &\quad + (\frac{3L^2\alpha_t}{2b_t m} - \frac{\alpha_t^2 L^3}{4m^2 b_t})\mathbb{E}[\|y_k^t - x_t\|^2] + (\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t} \\ &\leq \mathbb{E}[F(y_k^t)] - (\frac{m}{2\alpha_t} - \frac{L}{4})\mathbb{E}(\frac{\|y_{k+1}^t - x_t\|_2^2}{1+p}) - (\frac{m\alpha_t}{4} - \frac{L\alpha_t^2}{8})\mathbb{E}[\|g_{Y,k}^t\|_2^2] \\ &\quad + (\frac{3L^2\alpha_t}{2b_t m} - \frac{\alpha_t^2 L^3}{4m^2 b_t} + \frac{m}{2\alpha_t p} - \frac{L}{4p})\mathbb{E}[\|y_k^t - x_t\|^2] + (\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t}\end{aligned} \tag{52}$$

Let $p = 2k - 1$ and take summation with respect to the inner loop parameter k, we can get

$$\begin{aligned}&\mathbb{E}[F(x^{t+1})] \\ &\leq \mathbb{E}[F(x^t)] - \sum_{k=1}^{K}(\frac{m}{2\alpha_t(2k)} - \frac{L}{4(2k)})\mathbb{E}(\|y_{k+1}^t - x_t\|_2^2) - \sum_{k=1}^{K}(\frac{m\alpha_t}{4} - \frac{L\alpha_t^2}{8})\mathbb{E}[\|g_{Y,k}^t\|_2^2] \\ &\quad + \sum_{k=1}^{K}(\frac{3L^2\alpha_t}{2b_t m} - \frac{\alpha_t^2 L^3}{4m^2 b_t} + \frac{m}{2\alpha_t(2k-1)} - \frac{L}{4(2k-1)})\mathbb{E}[\|y_k^t - x_t\|^2] + \sum_{k=1}^{K}(\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t} \\ &\leq \mathbb{E}[F(x^t)] - \sum_{k=1}^{K-1}(\frac{m}{2\alpha_t(2k)} - \frac{L}{4(2k)})\mathbb{E}(\|y_{k+1}^t - x_t\|_2^2) - \sum_{k=1}^{K}(\frac{m\alpha_t}{4} - \frac{L\alpha_t^2}{8})\mathbb{E}[\|g_{Y,k}^t\|_2^2] \\ &\quad + \sum_{k=2}^{K}(\frac{3L^2\alpha_t}{2b_t m} - \frac{\alpha_t^2 L^3}{4m^2 b_t} + \frac{m}{2\alpha_t(2k-1)} - \frac{L}{4(2k-1)})\mathbb{E}[\|y_k^t - x_t\|^2] + \sum_{k=1}^{K}(\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t} \\ &\leq \mathbb{E}[F(x^t)] - \sum_{k=1}^{K}(\frac{m\alpha_t}{4} - \frac{L\alpha_t^2}{8})\mathbb{E}[\|g_{Y,k}^t\|_2^2] + \sum_{k=1}^{K}(\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t} \\ &\quad + \sum_{k=1}^{K-1}(\frac{3L^2\alpha_t}{2b_t m} - \frac{\alpha_t^2 L^3}{4m^2 b_t} + \frac{m}{2\alpha_t(2k+1)} - \frac{L}{4(2k+1)} - (\frac{m}{2\alpha_t(2k)} - \frac{L}{4(2k)}))\mathbb{E}[\|y_k^t - x_t\|^2] \\ &= \mathbb{E}[F(x^t)] - \sum_{k=1}^{K}(\frac{m\alpha_t}{4} - \frac{L\alpha_t^2}{8})\mathbb{E}[\|g_{Y,k}^t\|_2^2] + \sum_{k=1}^{K}(\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t} \\ &\quad + \sum_{k=1}^{K-1}(\frac{3L^2\alpha_t}{2b_t m} - \frac{\alpha_t^2 L^3}{4m^2 b_t} + (\frac{L}{4} - \frac{m}{2\alpha_t})(\frac{1}{2k(2k+1)}))\mathbb{E}[\|y_k^t - x_t\|^2]\end{aligned} \tag{53}$$

where the second inequality is due to the fact that $x_t = y_1^t$ and $\|x_{t+1} - x_t\| > 0$. Take $\alpha_t = m/L$

$$\begin{aligned}\mathbb{E}[F(x^{t+1})] &\leq \mathbb{E}[F(x^t)] - \sum_{k=1}^{K}\frac{m^2}{8L}\mathbb{E}[\|g_{Y,k}^t\|_2^2] + \sum_{k=1}^{K}(\frac{5}{4L})\frac{I(B_t<n)\sigma^2}{B_t} + \sum_{k=1}^{K-1}(\frac{5L}{4b_t} - \frac{L}{8k(2k+1)})\mathbb{E}[\|y_k^t - x_t\|^2] \\ &\leq \mathbb{E}[F(x^t)] - \sum_{k=1}^{K}\frac{m^2}{8L}\mathbb{E}[\|g_{Y,k}^t\|_2^2] + \sum_{k=1}^{K}(\frac{5}{4L})\frac{I(B_t<n)\sigma^2}{B_t}\end{aligned} \tag{54}$$

where the last inequality follows from the setting $K \leq \left\lfloor \sqrt{b_t/20} \right\rfloor$ and therefore

$$\frac{5L}{4b_t} - \frac{L}{8(K-1)(2(K-1)+1)} \leq \frac{5L}{4b_t} - \frac{L}{16K^2} \leq 0 \tag{55}$$

Take sum with respect to the outer loop parameter t and re-arrange the inequality

$$\begin{aligned}\sum_{t=1}^{T}\sum_{k=1}^{K}\frac{m^2}{8L}\mathbb{E}[\|g_{Y,k}^t\|_2^2] &\leq \mathbb{E}[F(x^1) - F(x^{T+1})] + \sum_{t=1}^{T}\sum_{k=1}^{K}(\frac{5}{4L})\frac{I(B_t<n)\sigma^2}{B_t} \\ &\leq \Delta_F + TK(\frac{5}{4L})\frac{I(B_t<n)\sigma^2}{B_t}\end{aligned} \tag{56}$$

Therefore when taking $B_t = n \wedge 20\sigma^2/(m^2\epsilon^2)$, $T = 1 \vee 16\Delta_F L/(m^2\epsilon^2 K)$

$$\mathbb{E}[\|g_{X,t^*}\|_2^2] \leq \frac{8\Delta_F L}{m^2 TK} + \frac{10I(B_t<n)\sigma^2}{B_t m^2} \leq \frac{\epsilon^2}{2} + \frac{\epsilon^2}{2} \leq \epsilon^2 \tag{57}$$

The total number of stochastic gradient computations is

$$\begin{aligned}TB + TKb &= O((n \wedge \frac{\sigma^2}{\epsilon^2} + b\sqrt{b})(1 + \frac{1}{\epsilon^2\sqrt{b}})) \\ &= O(n \wedge \frac{\sigma^2}{\epsilon^2} + b\sqrt{b} + \frac{n}{\epsilon^2\sqrt{b}} \wedge \frac{\sigma^2}{\epsilon^4\sqrt{b}} + \frac{b}{\epsilon^2}) \\ &= O(\frac{n}{\epsilon^2\sqrt{b}} \wedge \frac{\sigma^2}{\epsilon^4\sqrt{b}} + \frac{b}{\epsilon^2})\end{aligned} \tag{58}$$

where the last inequality is because $b^2 \leq \epsilon^{-4}$ when $b \leq \epsilon^{-2}$ and $\sqrt{b} \leq \epsilon^{-2}$ when ϵ^{-4}. However,we will never let b to be as large as ϵ^{-4} as it is even larger than the batch size B_t and doing so will make the number of gradient computations $O(\epsilon^{-6})$, which is undesirable.

Appendix C. Convergence under the PL Condition

C.1. Convergence under the PL condition

Now we provide the convergence of Algorithm 2 and 1 under the Polyak-Lojasiewicz(PL) condition [21]. Because of the existence of $h(x)$, we utilize the definition of the generalized PL condtion in Li and Li [18] to show the linear convergence rate of our generalized mirror descent algorithm with variance reduction under the condition. The generalized PL condition is defined as

$$\exists \mu > 0, s.t. \|g_{X,t}\|^2 \geq 2\mu(F(x_t) - F(x^*)), \forall x_t \tag{59}$$

where $g_{X,t}$ is the non-stochastic generalized gradient defined as in (2). Similar to Reddi et al. [22] and Li and Li [18], we assume the condition $L/(m^2\mu) \geq \sqrt{n}$ for simplicity. The condition is assumed only because we want to use same step size $\alpha_t = m/L$ as in Theorem 1 and if it is not satisfied, we can simply use a more complicated step size setting as in Li and Li [18]. We first provide the convergence result of Algorithm 2.

Theorem 10 *Suppose that f satisfies the Lipschitz gradients and bounded variance assumptions **A2, A3** and $\psi_{tk}(x)$ satisfy the m-strong convexity assumption **A1**. Further assume that the PL condition (59) is satisfied. The learning rate, the batch sizes, the mini-batch sizes, the number of inner loop iterations are set to be $\alpha_t = m/L, b_t = n \wedge (2(1+m^2)\sigma^2/(\epsilon m^2\mu))$. Then the output of algorithm 2 converges with gradient computations*

$$O((\frac{n}{\mu} \wedge \frac{\sigma^2}{\mu^2\epsilon}) \log \frac{1}{\epsilon}) \tag{60}$$

Remark. The proof is relegated to Appendix A. The above result is $\tilde{O}(n\mu^{-1} \wedge \mu^{-2}\epsilon^{-1})$ when we hide logarithm terms and treat σ^2 as a constant. Similar to Theorem 3, the SFO complexity matches the smaller complexity of ProxSGD and ProxGD under the PL condition [13].

Next we present the convergence of the variance reduced Algorithm 1.

Theorem 11 *Suppose that f satisfies the Lipschitz gradients and bounded variance assumptions **A2, A3** and $\psi_{tk}(x)$ satisfy the m-strong convexity assumption **A1**. Further assume that the PL condition (59) is satisfied. The learning rate, the batch sizes, the mini-batch sizes, the number of inner loop iterations are set to be $\alpha_t = m/L, B_t = n \wedge (10\sigma^2/(\epsilon m^2\mu)), b_t = b, K = \lfloor\sqrt{b/32}\rfloor \vee 1$. Then the output of algorithm 1 converges with gradient computations*

$$O((n \wedge \frac{\sigma^2}{\mu\epsilon})\frac{1}{\mu\sqrt{b}} \log \frac{1}{\epsilon} + \frac{b}{\mu} \log \frac{1}{\epsilon}) \tag{61}$$

Remark. The proof is relegated to Appendix C. The above result is $\tilde{O}((n \wedge (\mu\epsilon)^{-1})(\mu\sqrt{b})^{-1} + b\mu^{-1})$ when we hide logarithm terms and treat σ^2 as a constant. Similar to results in Theorem 1, the gradient complexity is asymptotically the same as ProxSVRG+, as shown in Table 1. Compared with the complexity of Algorithm 2, our complexity can be arguably better when we choose appropriate mini batch sizes b, which further proves our conclusion that variance reduction can be applied to any adaptive SMD algorithm to reduce the gradient complexity. We provide the following corollary for one choice of b to show its effectiveness.

Corollary 12 *With all the assumptions and parameter settings in Theorem 11, further assume that $b = (\mu\epsilon)^{-2/3}$. Then the output of algorithm 1 converges with gradient computations*

$$O((\frac{n\epsilon^{1/3}}{\mu^{2/3}} \wedge \frac{\epsilon^{-2/3}}{\mu^{5/3}} + \frac{\epsilon^{-2/3}}{\mu^{5/3}}) \log \frac{1}{\epsilon}) \tag{62}$$

Remark. The above complexity is the same as the best complexity of ProxSVRG+, with the same choice of mini-batch sizes b. Moreover, it generalizes the best results of ProxSVRG/ProxSAGA and SCSG, without the need to perform any restarts as in ProxSVRG [22]. Therefore, as ProxSVRG+, it automatically switch to fast convergence in regions satisfying the PL condition.

C.2. Proof. of Convergence under the PL condition

Recall the definition of the PL condition and modify the notations a little bit, we get

$$\exists \mu > 0, s.t. \|g_{Y,k}^t\|^2 \geq 2\mu(F(y_k^t) - F(x^*)) \tag{63}$$

By the proof in appendix B, we know that

$$\begin{aligned}
\mathbb{E}[F(y_{k+1}^t)] &\leq \mathbb{E}[F(y_k^t)] - (\frac{m}{2\alpha_t} - \frac{L}{4})\mathbb{E}(\frac{\|y_{k+1}^t - x_t\|_2^2}{1+p} - \frac{\|y_k^t - x_t\|_2^2}{p}) - (\frac{m\alpha_t}{4} - \frac{L\alpha_t^2}{8})\mathbb{E}[\|g_{Y,k}^t\|_2^2] \\
&\quad + (\frac{3L^2\alpha_t}{2b_t m} - \frac{\alpha_t^2 L^3}{4m^2 b_t})\mathbb{E}[\|y_k^t - x_t\|^2] + (\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t} \\
&\leq \mathbb{E}[F(y_k^t)] - (\frac{m}{2\alpha_t} - \frac{L}{4})\mathbb{E}(\frac{\|y_{k+1}^t - x_t\|_2^2}{1+p}) - (\frac{m\alpha_t}{2} - \frac{L\alpha_t^2}{4})\mu(\mathbb{E}[F(y_k^t)] - F(x^*)) \\
&\quad + (\frac{3L^2\alpha_t}{2b_t m} - \frac{\alpha_t^2 L^3}{4m^2 b_t} + \frac{m}{2\alpha_t p} - \frac{L}{4p})\mathbb{E}[\|y_k^t - x_t\|^2] + (\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t}
\end{aligned} \tag{64}$$

Therefore when $p = 2k-1$, define $\gamma := (1 - (\frac{m\alpha_t\mu}{2} - \frac{L\alpha_t^2\mu}{4}))$, we obtain

$$\begin{aligned}
\frac{\mathbb{E}[F(y_{k+1}^t)] - F(x^*)}{\gamma^{k+1}} &\leq \frac{(\mathbb{E}[F(y_k^t)] - F(x^*))}{\gamma^k} - (\frac{m}{2\alpha_t\gamma^{k+1}} - \frac{L}{4\gamma^{k+1}})\mathbb{E}(\frac{\|y_{k+1}^t - x_t\|_2^2}{2k}) \\
&\quad + \frac{1}{\gamma^{k+1}}(\frac{3L^2\alpha_t}{2b_t m} - \frac{\alpha_t^2 L^3}{4m^2 b_t} + \frac{m}{2\alpha_t(2k-1)} - \frac{L}{4(2k-1)})\mathbb{E}[\|y_k^t - x_t\|^2] \\
&\quad + \frac{1}{\gamma^{k+1}}(\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t}
\end{aligned} \tag{65}$$

Summing up with respect to the inner loop parameter k

$$\begin{aligned}
\mathbb{E}[F(x_{t+1})] - F(x^*) &\leq \gamma^K(\mathbb{E}[F(x_t)] - F(x^*)) - \gamma^{K+1}\sum_{k=1}^K(\frac{m}{2\alpha_t\gamma^{k+1}} - \frac{L}{4\gamma^{k+1}})\mathbb{E}(\frac{\|y_{k+1}^t - x_t\|_2^2}{2k}) \\
&\quad + \gamma^{K+1}\sum_{k=1}^K\frac{1}{\gamma^{k+1}}(\frac{3L^2\alpha_t}{2b_t m} - \frac{\alpha_t^2 L^3}{4m^2 b_t} + \frac{m}{2\alpha_t(2k-1)} - \frac{L}{4(2k-1)})\mathbb{E}[\|y_k^t - x_t\|^2] \\
&\quad + \gamma^{K+1}\sum_{k=1}^K\frac{1}{\gamma^{k+1}}(\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t} \\
&= \gamma^K(\mathbb{E}[F(x_t)] - F(x^*)) - \gamma^{K+1}\sum_{k=1}^K(\frac{m}{2\alpha_t\gamma^{k+1}} - \frac{L}{4\gamma^{k+1}})\mathbb{E}(\frac{\|y_{k+1}^t - x_t\|_2^2}{2k}) \\
&\quad + \gamma^{K+1}\sum_{k=1}^K\frac{1}{\gamma^{k+1}}(\frac{3L^2\alpha_t}{2b_t m} - \frac{\alpha_t^2 L^3}{4m^2 b_t} + \frac{m}{2\alpha_t(2k-1)} - \frac{L}{4(2k-1)})\mathbb{E}[\|y_k^t - x_t\|^2] \\
&\quad + \frac{1-\gamma^K}{1-\gamma}(\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t}
\end{aligned} \tag{66}$$

By the fact that $x_t = x_1^t$, and $\|x_{t+1} - x_t\| > 0$, we know that

$$
\begin{aligned}
&\mathbb{E}[F(x_{t+1})] - F(x^*) \\
&\leq \gamma^K(\mathbb{E}[F(x_t)] - F(x^*)) - \gamma^{K+1}\sum_{k=1}^{K-1}(\frac{m}{2\alpha_t\gamma^{k+1}} - \frac{L}{4\gamma^{k+1}})\mathbb{E}(\frac{\|y_{k+1}^t - x_t\|_2^2}{2k}) \\
&\quad + \gamma^{K+1}\sum_{k=2}^{K}\frac{1}{\gamma^{k+1}}(\frac{3L^2\alpha_t}{2b_t m} - \frac{\alpha_t^2 L^3}{4m^2 b_t} + \frac{m}{2\alpha_t(2k-1)} - \frac{L}{4(2k-1)})\mathbb{E}[\|y_k^t - x_t\|^2] \\
&\quad + \frac{1-\gamma^K}{1-\gamma}(\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t} \\
&= \gamma^K(\mathbb{E}[F(x_t)] - F(x^*)) + \frac{1-\gamma^K}{1-\gamma}(\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t} \\
&\quad - \gamma^{K+1}\sum_{k=1}^{K-1}(\frac{m}{2\alpha_t\gamma^{k+1}} - \frac{L}{4\gamma^{k+1}})\mathbb{E}(\frac{\|y_{k+1}^t - x_t\|_2^2}{2k}) \\
&\quad + \gamma^{K+1}\sum_{k=1}^{K-1}\frac{1}{\gamma^{k+1}}(\frac{3L^2\alpha_t}{2b_t m\gamma} - \frac{\alpha_t^2 L^3}{4m^2 b_t\gamma} + \frac{m}{2\alpha_t(2k+1)\gamma} - \frac{L}{4(2k+1)\gamma})\mathbb{E}[\|y_k^t - x_t\|^2] \\
&= \gamma^K(\mathbb{E}[F(x_t)] - F(x^*)) + \frac{1-\gamma^K}{1-\gamma}(\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t} \\
&\quad + \gamma^{K+1}\sum_{k=1}^{K-1}\frac{1}{\gamma^{k+2}}(\frac{3L^2\alpha_t}{2b_t m} - \frac{\alpha_t^2 L^3}{4m^2 b_t} + \frac{m}{2\alpha_t(2k+1)} - \frac{L}{4(2k+1)} + \frac{L\gamma}{8k} - \frac{m\gamma}{4k\alpha_t})\mathbb{E}[\|y_k^t - x_t\|^2] \\
&= \gamma^K(\mathbb{E}[F(x_t)] - F(x^*)) + \frac{1-\gamma^K}{1-\gamma}(\frac{3\alpha_t}{2m} - \frac{\alpha_t^2 L}{4m^2})\frac{I(B_t < n)\sigma^2}{B_t} \\
&\quad + \gamma^{K+1}\sum_{k=1}^{K-1}\frac{1}{\gamma^{k+2}}(\frac{3L^2\alpha_t}{2b_t m} - \frac{\alpha_t^2 L^3}{4m^2 b_t} - (\frac{m}{2\alpha_t} - \frac{L}{4})(\frac{\gamma}{2k} - \frac{1}{2k+1}))\mathbb{E}[\|y_k^t - x_t\|^2]
\end{aligned}
\tag{67}
$$

By the definition $\gamma = 1 - \frac{m\alpha_t\mu}{2} + \frac{L\alpha_t^2\mu}{4}$, we know that

$$
\begin{aligned}
\frac{\gamma}{2k} - \frac{1}{2k+1} &= \frac{1}{2k(2k+1)} - \frac{m\alpha_t\mu}{4k} + \frac{L\alpha_t^2\mu}{8k} \\
&= \frac{1}{2k(2k+1)} - \frac{\alpha_t^2\mu}{2k}(\frac{m}{2\alpha_t} - \frac{L}{4})
\end{aligned}
\tag{68}
$$

Therefore when taking $\alpha_t = \frac{m}{L}$ and with the assumption $L/(\mu m^2) > \sqrt{n}$, the last term in the inequality (67) is

$$
\begin{aligned}
&\gamma^{K+1}\sum_{k=1}^{K-1}\frac{1}{\gamma^{k+2}}(\frac{3L^2\alpha_t}{2b_tm}-\frac{\alpha_t^2L^3}{4m^2b_t}-(\frac{m}{2\alpha_t}-\frac{L}{4})(\frac{\gamma}{2k}-\frac{1}{2k+1})\mathbb{E}[\|y_k^t-x_t\|^2]\\
&=\gamma^{K+1}\sum_{k=1}^{K-1}\frac{1}{\gamma^{k+2}}(\frac{3L^2\alpha_t}{2b_tm}-\frac{\alpha_t^2L^3}{4m^2b_t}-(\frac{m}{2\alpha_t}-\frac{L}{4})(\frac{1}{2k(2k+1)}-\frac{\alpha_t^2\mu}{2k}(\frac{m}{2\alpha_t}-\frac{L}{4})))\mathbb{E}[\|y_k^t-x_t\|^2]\\
&=\gamma^{K+1}\sum_{k=1}^{K-1}\frac{1}{\gamma^{k+2}}(\frac{3L^2\alpha_t}{2b_tm}-\frac{\alpha_t^2L^3}{4m^2b_t}-(\frac{m}{2\alpha_t}-\frac{L}{4})(\frac{1}{2k(2k+1)})+\frac{\alpha_t^2\mu}{2k}(\frac{m}{2\alpha_t}-\frac{L}{4})^2))\mathbb{E}[\|y_k^t-x_t\|^2]\\
&\leq\gamma^{K+1}\sum_{k=1}^{K-1}\frac{1}{\gamma^{k+2}}(\frac{5L}{4b_t}-\frac{L}{4}(\frac{1}{2k(2k+1)})+\frac{L}{32k\sqrt{n}})\mathbb{E}[\|y_k^t-x_t\|^2]
\end{aligned}
\tag{69}
$$

Define $H(x):=-\frac{1}{2x(2x+1)}+\frac{1}{8x\sqrt{n}}+\frac{5}{b_t}$, $H'(x)=\frac{8x+2}{4x^2(2x+1)^2}-\frac{1}{8x^2\sqrt{n}}=\frac{1}{4x^2}(\frac{8x+2}{4x^2+4x+1}-\frac{1}{2\sqrt{n}})=\frac{1}{4x^2}(\frac{2(8x+2)\sqrt{n}-(4x^2+4x+1)}{2(4x^2+4x+1)\sqrt{n}})$. When $x\leq K-1<K<\sqrt{\frac{b_t}{16}}\leq\sqrt{\frac{n}{16}}$, $\frac{8x+2}{4x^2+4x+1}-\frac{1}{2\sqrt{n}}\geq\frac{8K+2}{4K^2+4K+1}-\frac{1}{2\sqrt{n}}\geq 0$. Therefore $H(x)\leq H(K-1)\leq\frac{5}{b_t}-\frac{14K+1}{32K(K-1)(2K-1)}\leq 0$ when $K=\lfloor\sqrt{\frac{b_t}{32}}\rfloor$. which means the inequality above is smaller than zero. Hence

$$
\mathbb{E}[F(x_{t+1})]-F(x^*)\leq\gamma^K(\mathbb{E}[F(x_t)]-F(x^*))+\frac{1-\gamma^K}{1-\gamma}\frac{5L}{4}\frac{I(B_t<n)\sigma^2}{B_t}
\tag{70}
$$

Therefore

$$
\frac{\mathbb{E}[F(x_{t+1})]-F(x_t)}{\gamma^{K(t+1)}}\leq\frac{(\mathbb{E}[F(x_t)]-F(x^*))}{\gamma^{Kt}}+\frac{1-\gamma^K}{(1-\gamma)\gamma^{K(t+1)}}(\frac{5L}{4}\frac{I(B_t<n)\sigma^2}{B_t})
\tag{71}
$$

Now take sum with respect to the outer loop parameter t and take B_t as a constant, we can get

$$
\begin{aligned}
\mathbb{E}[F(x_{T+1})]-F(x^*)&\leq\gamma^{KT}(F(x_1)-F(x^*))+\gamma^{K(T+1)}\sum_{t=1}^{T}\frac{1-\gamma^K}{(1-\gamma)\gamma^{K(t+1)}}\frac{5L}{4}\frac{I(B_t<n)\sigma^2}{B_t}\\
&\leq\gamma^{KT}\Delta_F+\gamma^{K(T+1)}\frac{1-\gamma^K}{1-\gamma}\sum_{t=1}^{T}\frac{1}{\gamma^{K(t+1)}}\frac{5L}{4}\frac{I(B_t<n)\sigma^2}{B_t}\\
&=\gamma^{KT}\Delta_F+\gamma^{K(T+1)}\frac{1-\gamma^K}{1-\gamma}\sum_{t=1}^{T}\frac{1}{\gamma^{K(t+1)}}\frac{5L}{4}\frac{I(B_t<n)\sigma^2}{B_t}\\
&=\gamma^{KT}\Delta_F+\frac{5LI(B_t<n)\sigma^2}{4B_t}\frac{1-\gamma^K}{1-\gamma}\frac{1-\gamma^{KT}}{1-\gamma^K}\\
&=\gamma^{KT}\Delta_F+\frac{5LI(B_t<n)\sigma^2}{4B_t}\frac{1-\gamma^{KT}}{1-\gamma}
\end{aligned}
\tag{72}
$$

Since $1-\gamma^{KT}<1$, $\gamma=1-\frac{m\alpha_t\mu}{2}+\frac{L\alpha_t^2\mu}{4}=1-\frac{m^2\mu}{2L}+\frac{m^2\mu}{4L}=1-\frac{m^2\mu}{4L}$, hence

$$\begin{aligned}\mathbb{E}[F(x_{T+1})] - F(x^*) &\leq \gamma^{KT}\Delta_F + \frac{5LI(B_t < n)\sigma^2}{4B_t(1-\gamma)} \\ &= \gamma^{KT}\Delta_F + \frac{5I(B_t < n)\sigma^2}{B_t m^2 \mu}\end{aligned} \tag{73}$$

Therefore when taking $T = 1 \vee (\log \frac{2\Delta_F}{\epsilon})/(K \log \frac{1}{\gamma}) = O((\log \frac{2\Delta_F}{\epsilon})/(K\mu))$, $B_t = n \wedge \frac{10\sigma^2}{\epsilon m^2 \mu}$. Then the total number of stochastic gradient computations is

$$\begin{aligned}TB + TKb &= O((n \wedge \frac{\sigma^2}{\mu\epsilon} + b\sqrt{b})(\frac{1}{\mu\sqrt{b}} \log \frac{1}{\epsilon})) \\ &= O((n \wedge \frac{\sigma^2}{\mu\epsilon})\frac{1}{\mu\sqrt{b}} \log \frac{1}{\epsilon} + \frac{b}{\mu} \log \frac{1}{\epsilon})\end{aligned} \tag{74}$$

Appendix D. Algorithm Implementation and More Experimental Details

Datasets. We used two datasets in our experiments. The MNIST [23] dataset has 50k training images and 10k testing images of handwritten digits. The images were normalized before fitting into the neural networks. The CIFAR10 dataset [15] also has 50k training images and 10k testing images of different objects in 10 classes. The images were normalized with respect to each channel (3 channels in total) before fitting into the network.

Network Architecture. For the MNIST dataset, we used a one-hidden layer fully connected neural network as the architecture. The hidden layer size was 64 and we used the Relu activation function [20]. The logsoftmax activation function was applied to the final output. For CIFAR-10, we used the standard LeNet [16] with two layers of convolutions of size 5. The two layers have 6 and 16 channels respectively. Relu activation and max pooling are applied to the output of each convolutional layer. The output is then applied sequentially to three fully connected layers of size 120, 84 and 10 with Relu activation functions.

Implementations and Parameter Tuning. All experiments are conducted independently on NVIDIA Tesla P100 GPUs. Except for normalization, we did not perform any additional data transformation or augmentation techniques such as rotation, flipping, and cropping on the images, which was the same as what Zhou et al. [27] did in their experiments. For the constant m added to the denominator matrix H_t in these two algorithms, we choose a reasonable value of $m = 0.001$, which is common in real implementations [14]. The other parameters are set to be the default values. For example, the exponential moving average parameter β in RMSProp is set to be 0.999. For the step sizes α_t, we tuned over $\{0.1, 0.01, 0.005, 0.002, 0.001\}$ for all the algorithms. For the mini batch sizes of AdaGrad and RMSProp, we tuned over $\{256, 512, 1024, 2048, 4096\}$. For the batch sizes and mini batch sizes B_t and b_t in VR-AdaGrad, VR-RMSProp, we used a slightly different notation of batch size ratio $r = B_t/b_t$. We tuned over $b_t = \{64, 128, 256, 512, 1024\}$ and $r = \{4, 8, 16, 32, 64\}$ and reported the best results for each algorithm on each dataset. The parameters that generated the reported results were provided in Table 2 and 3 in Appendix D. No step size decay was applied to any algorithms in our experiments. However, according to our theory, step size decay would not affect our conclusions since the step sizes were upper bounded.

We provide the implementation of Variance Reduced AdaGrad (VR-AdaGrad) in Algorithm 4. Note that this implementation is actually a simple combination of the AdaGrad algorithm and

Algorithm 4 AdaGrad with Variance Reduction Algorithm

1: **Input:** Number of stages T, initial x_1, step sizes $\{\alpha_t\}_{t=1}^T$, batch sizes $\{B_t\}_{t=1}^T$, mini-batch sizes $\{b_t\}_{t=1}^T$, constant m
2: **for** $t = 1$ **to** T **do**
3: Randomly sample a batch $\mathcal{I}_t$ with size B_t
4: $g_t = \nabla f_{\mathcal{I}_t}(x_t)$
5: $y_1^t = x_t$
6: **for** $k = 1$ **to** K **do**
7: Randomly pick sample $\tilde{\mathcal{I}}_t$ of size b_t
8: $v_k^t = \nabla f_{\tilde{\mathcal{I}}_t}(y_k^t) - \nabla f_{\tilde{\mathcal{I}}_t}(y_1^t) + g_t$
9: $y_{k+1}^t = y_k^t - \alpha_t v_k^t / (\sqrt{\frac{1}{(t-1)K+k}(\sum_{t=1}^{t-1}\sum_{i=1}^{K} v_k^{t2} + \sum_{i=1}^{k} v_k^{t2})} + m)$
10: **end for**
11: $x_{t+1} = y_{K+1}^t$
12: **end for**
13: **Return** (Smooth case) Uniformly sample x_{t^*} from $\{y_k^t\}_{k=1,t=1}^{K,T}$; (P-L case) $x_{t^*} = x_{T+1}$

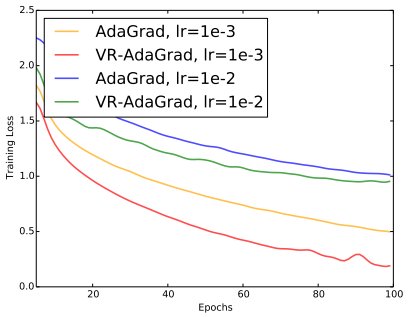
the SVRG algorithm with $h(x) = 0$. The implementation can be further extended to the case when $h(x) \neq 0$, but the form would depend on the regularization function $h(x)$. For example, the AdaGrad algorithm with $h(x) = \|x\|_1$, a non-smooth regularization, can be found in Duchi et al. [7]. The VR-AdaGrad algorithm with $h(x) = \|x\|_1$ will therefore have a similar form. To change the algorithm into VR-RMSProp, one can simply replace the global average design of the denominator with the exponential moving average in line 9.

The parameter settings that we used to generate the best results in our section 4 are reported in Table 2 and Table 3. The tuning details are presented in section 4. Note that for variance reduced AdaGrad (VR-AdaGrad) and variance reduced RMSProp (VR-RMSProp), we have two parameters B_t and b_t. To compute the batch size B_t, we simply need to multiply b_t by r.

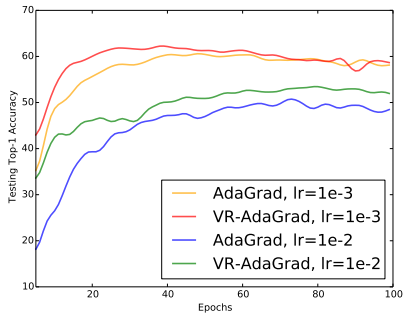
We provide the performances of AdaGrad, RMSProp and their variance reduced variants with different step sizes in figure 3 and 4. Note that variance reduction always works in these figures, and it results in faster convergence and better testing accuracy. Therefore for different step sizes, we can always apply variance reduction to get faster training and better performances.

Table 2: Best parameter settings on the MNIST dataset. The batch size B_t is equal to $b_t * r$.

Algorithms	Step size	Mini batch size b_t	Batch size ratio r
SGD	0.01	1024	N.A.
AdaGrad	0.001	2048	N.A.
RMSProp	0.001	1024	N.A.
ProxSVRG+	0.01	256	32
VR-AdaGrad	0.001	256	32
VR-RMSProp	0.001	256	64

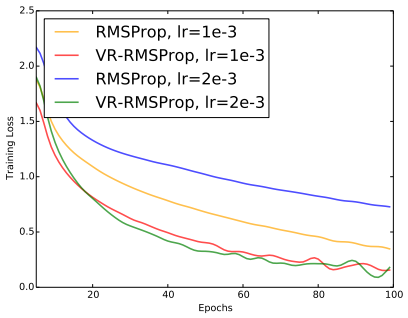


(a) CIFAR-10 Training Loss

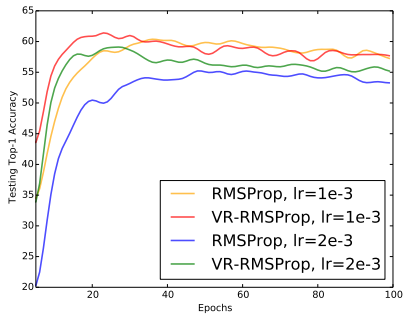


(b) CIFAR-10 Testing Acc.

Figure 3: Comparison of AdaGrad and VR-AdaGrad on CIFAR-10 using different learning rates. The other parameters are the same as in Table 3. "lr" stands for learning rate, which is a different name for step size. The results are averaged over 5 independent runs



(a) CIFAR-10 Training Loss

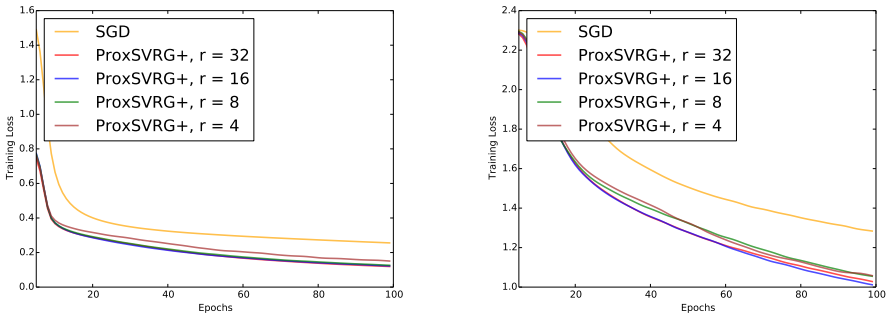


(b) CIFAR-10 Testing Acc.

Figure 4: Comparison of RMSProp and VR-RMSProp on CIFAR-10 using different learning rates. The other parameters are the same as in Table 3. "lr" stands for learning rate, which is a different name for step size. lr=1e-2 is too large for RMSProp and the algorithm diverges. The results are averaged over 5 independent runs

Table 3: Best parameter settings on the CIFAR-10 dataset. The batch size B_t is equal to $b_t * r$.

Algorithms	Step size	Mini batch size b_t	Batch size ratio r
SGD	0.01	1024	N.A.
AdaGrad	0.001	1024	N.A.
RMSProp	0.001	1024	N.A.
ProxSVRG+	0.01	512	32
VR-AdaGrad	0.001	512	32
VR-RMSProp	0.001	512	64



(a) ProxSVRG+ on MNIST (b) ProxSVRG+ on CIFAR-10

Figure 5: 5(a)subfigure training loss of SGD and ProxSVRG+ with different r on MNIST. 5(b)subfigure training loss of SGD and ProxSVRG+ with different r on CIFAR-10. The other parameters are the same as in Table 2, 3. The mini batch size is set to be the same as AdaGrad and RMSProp to ensure fair comparisons. The results were averaged over three independent runs.