Journal of Machine Learning Research 25 (2024) 1-87 Submitted 10/22; Revised 8/24; Published 10/24

GGD: Grafting Gradient Descent

Yanjing Feng YJFENG@MAIL.NANKAI.EDU.CN
NITFID, School of Statistics and Data science
Nankai University
Tianjin 300071, China

Yongdao Zhou YDZHOU@NANKAI.EDU.CN
NITFID, School of Statistics and Data science
Nankai University
Tianjin 300071, China

Editor: Moritz Hardt

Abstract

Simple random sampling has been widely used in traditional stochastic optimization algorithms. Although the gradient sampled by simple random sampling is a descent direction in expectation, it may have a relatively high variance which will cause the descent curve wiggling and slow down the optimization process. In this paper, we propose a novel stochastic optimization method called grafting gradient descent (GGD), which combines the strength from minibatching and importance sampling, and provide the convergence results of GGD. We show that the grafting gradient possesses a doubly robust property which ensures that the performance of GGD method is superior to the worse one of SGD with importance sampling method and mini-batch SGD method. Combined with advanced variance reduction techniques such as stochastic variance reduced gradient and adaptive stepsize methods such as Adam, these composite GGD-based methods and their theoretical bounds are provided. The real data studies also show that GGD achieves an intermediate performance among SGD with importance sampling and mini-batch SGD, and outperforms original SGD method. Then the proposed GGD is a better and more robust stochastic optimization framework in practice.

Keywords: stochastic optimization, importance sampling, minibatching, variance reduction, adaptive stepsize method

1. Introduction

One fundamental problem studied in machine learning is how to fit the model to large data set. The most popular approach is via the empirical risk minimization (ERM), that is,

$$\min_{x\in\mathbb{R}^d}\left\{f(x)=\frac{1}{n}\sum_{i=1}^{n}f_i(x)\right\},$$

where x is the d parameters in a pre-defined model, and $f_i(x)$ is the loss function of the sample i, $i=1,2,...,n$, such as the square error loss or hinge error loss. Optimizing the objective function f is of paramount importance. The most well-known method is via the stochastic gradient descent, whose update rule for the model parameters x can be written

as

$$x^{k+1} = x^k - \gamma \nabla f_{i_k}(x^k),$$

where i_k is sampled from $[n] = \{1, 2, ...n\}$ uniformly, γ is a suitable stepsize, $\nabla f_{i_k}(x^k) = \left(\frac{\partial f_{i_k}}{\partial x_1}(x^k), ..., \frac{\partial f_{i_k}}{\partial x_d}(x^k)\right)^\top$. SGD has played a central role in large-scale machine learning (Robbins and Monro, 1951; Shalev-Shwartz et al., 2011; Hardt et al., 2016; Bottou et al., 2018; Gorbunov et al., 2020), since it tremendously reduces the computational cost compared with the gradient descent (GD). Unfortunately, SGD suffers from the variance brought by the sample gradient $\nabla f_{i_k}(x)$. In terms of practice, main problem with SGD is countering its variance to accelerate the training process.

1.1 Direct Approaches

Some techniques can be used to directly tackle the problem of variance. They mainly fall into three categories.

Minibatching. Minibatching can reduce the variance by a constant factor. Using this strategy does not result in an improvement of convergence rate (Shalev-Shwartz et al., 2011; Bottou et al., 2018), but can lead to acceleration. Minibatching is commonly used in modern deep learning settings since it can be running in a parallel fashion which is computational friendly for implementation.

Importance sampling. Importance sampling refers to the technique of assigning carefully designed non-uniform probabilities to data samples and using these probabilities to select the data point during the iterative training process (Needell et al., 2014; Zhao and Zhang, 2015; El Hanchi and Stephens, 2020). Although effective, it often requires more computational resources to set up the sampling mechanism. With the help of importance sampling, the variance of stochastic gradient can be reduced by a factor as well.

Variance reduction methods Although above two primitive techniques can be implemented easily and reduce the variance to some extent, they can not eliminate the variance completely. To improve the convergence rate of stochastic optimization methods with fixed stepsize, lots of advanced algorithms concerning the variance reduction have been proposed in recent years (Gower et al., 2020), such as stochastic average gradient (SAG, Le Roux et al., 2012; Schmidt et al., 2017), SAGA (Defazio et al., 2014), stochastic variance reduced gradient (SVRG, Johnson and Zhang, 2013), stochastic recursive gradient algorithm (SARAH, Nguyen et al., 2017), Katyusha (Allen-Zhu, 2017), variance reduced stochastic gradient descent (VR-SGD, Shang et al., 2018), and the simple stochastic variance reduced algorithm (MiG, Zhou et al., 2018). All of those methods have modified the original stochastic sample gradient $\nabla f_{i_k}(x)$ in each step to progressively reduce its variance as an estimator of the full gradient. Among them, SVRG first introduces the bi-loop structure where the parameters are updated in the inner loop and the reference point and the full gradient are updated in the outer loop. Compared with the concurrently proposed methods SAG and SAGA, SVRG does not require storing a Jacobian matrix in the training process and produces an unbiased estimator of the full gradient in each step. It is worth noting that another weakness of variance reduction methods is that they are inefficient and the reduction in variance is insignificant for deep learning (Defazio and Bottou, 2019).

Minibatching, importance sampling and advanced variance reduction methods are often combined to be used for achieving an amplification effect. Thought of combining these techniques is by no means new. We list some examples as follows.

Minibatching and importance sampling. The most natural way is to predefine a sampling probability of each data sample and uses a mini batch of sampled gradients (averaging multiple sampled gradients) to update the model parameters (Zhao and Zhang, 2015; Qian et al., 2019). Csiba and Richtárik (2018) propose importance sampling for mini-batches, which gives the answer to the problem *which subset should we choose in every iteration*. A key characteristic of those proposed methods is that they will define a sampling probability on the entire data set and then do the sampling step to accelerate the training process.

Minibatching and variance reduction. Reddi et al. (2016) give the experiment and convergence results for mini-batch nonconvex SVRG, and concurrently Allen-Zhu and Hazan (2016) provide the convergence results for non-convex SVRG and claim that their convergence results can be extended to the mini-batch setting. Yang et al. (2021) study the mini-batch SARAH with random Barzilai-Borwein method. Gazagnadou et al. (2019) study the optimal mini-batch size for SAGA. mS2GD proposed by Konečnỳ et al. (2015) is another example of this kind of hybrid.

Minibatching, importance sampling and variance reduction. Horváth and Richtárik (2019) pioneer the study of variance reduction method with minibatching and importance sampling in the nonconvex problem.

1.2 Indirect Approaches

Different from the methods which directly reduce the variance of stochastic gradient. Another way to accelerate the training process is tunning the stepsize. Diminishing stepsize sequences and adaptive stepsize methods are two representative examples of this kind. They are introduced as follows.

Diminishing stepsize sequences. Although using a diminishing stepsize sequence (Cotter et al., 2011) can eliminate the variance gradually through iterations, it also slows down the convergence rate of an algorithm as what we illustrate in Section 4. Moreover, the performance of SGD algorithm can be deteriorated by a wrongly hand-picked sequence. A suitable stepsize sequence can not be obtained without the trial and error.

Adaptive stepsize methods. Unlike the diminishing stepsize sequences which put the same stepsize on each dimension of the model parameters. Adaptive stepsize methods assign different stepsizes for each dimension of the model parameters and update them separately. Many stochastic first-order optimization methods with adaptive stepsize and momentum has been proven both theoretically and empirically that they can accelerate the training process such as SGD with momentum (Liu et al., 2020), RMSprop (Tieleman et al., 2012; Zou et al., 2019), Adadelta (Zeiler, 2012), Adagrad (Duchi et al., 2011; Ward et al., 2020), Adam (Kingma and Ba, 2014) and AMSgrad (Reddi et al., 2019; Tran et al., 2019). Among them, Kingma and Ba (2014) decide to use exponential moving average to cumulate the past gradients with heavy-ball style momentum which is used to determine the direction and the original magnitude of the update, and as well cumulate their element-wise square with the corrective terms to modify the original magnitude of the update so

that the element-wise second moment of the update can be regularized to approximate 1. The proposed Adam is now one of the most popular adaptive stepsize method used in deep learning community.

1.3 Our Contributions

Motivated by these composite methods, we propose a novel stochastic optimization method, grafting gradient descent (GGD), which borrows the strength from minibatching and importance sampling. Since stochastic sample gradient has intrinsic high variance, we replace the stochastic gradient with a brand new grafting gradient to update the model parameters, and integrate the grafting gradient with the high-level variance reduction methods and adaptive stepsize methods respectively. Our contributions of this work are as follows:

- Grafting gradient has a smaller noise variance compared with the stochastic sample gradient and can be calculated in a parallel fashion to speed up the training process.
- Two types of GGD methods are proposed, GGD using sampling with replacement and GGD using sampling without replacement. For the former one, we prove that the noise variance of grafting gradient can be written as the weighted sum of the noise variance of SGD with importance sampling and the noise variance of mini-batch SGD which guarantees grafting gradient a doubly robust estimator with respect to the full gradient.
- For the latter one, we show that vanilla SGD, SGD with importance sampling, mini-batch SGD can all be regarded as the special cases of GGD using sampling without replacement. A unified bound is obtained through the convergence analysis of GGD using sampling without replacement.
- Two types of GGD methods both have a sublinear rate of convergence under strongly-convex assumption when using a diminishing stepsize sequence and a linear convergence rate up to some noise level with the fixed stepsize. We also provide the convergence analysis for GGD in the general convex and non-convex cases.
- Grafting gradient is also compatible with advanced variance reduction method and adaptive stepsize method respectively. The convergence analysis of GGD-based variance reduction method and GGD-based adaptive stepsize method are provided and we show that these methods converge much faster than the original GGD-based methods.

The rest of this paper is organized as follows. Section 2 presents some definitions and notations for further convergence analysis. Section 3 introduces the grafting gradient and the detail of two type GGD algorithms. Section 4 gives the convergence results of GGD for strongly-convex, convex and non-convex objective function respectively. In Section 5, we hybridize SVRG and Adam with the grafting gradient using sampling with replacement (WR), introduce GGD-WR-SVRG and GGD-WR-Adam methods, and provide their theoretical properties. Section 6 gives the experimental results. Section 7 gives some conclusions and discussions for the future research. All the proofs are listed in the Appendix. All the codes are available at https://github.com/oo0mmmm/GGD.

2. Background and Problem Setup

We first introduce some notations which are repeatedly used in the rest of this paper. Let

$$f_S(x) = \frac{1}{|S|}\sum_{i\in S} f_i(x),\ L_S = \frac{1}{|S|}\sum_{i\in S} L_i, \text{ and } f_{S,min} = \frac{1}{|S|}\sum_{i\in S} f_{i,min},$$

where S is a subset of training set, L_i is the smoothness constant and $f_{i,min}$ is a lower bound of $f_i(x)$. To proceed with the convergence analysis, we also presents some basic definitions which are widely used in stochastic optimization as follows.

Definition 1 (L-smoothness) *Function f: $\mathbb{R}^d \to \mathbb{R}$, is L-smooth if it is continuously differentiable and the gradient function of f, namely, $\nabla f : \mathbb{R}^d \to \mathbb{R}^d$, is Lipschitz continuous with Lipschitz constant $L > 0$, i.e.,*

$$\|\nabla f_i(x) - \nabla f_i(\bar{x})\|_2 \leq L\|x - \bar{x}\|_2, \textit{ for all } (x, \bar{x}) \in \mathbb{R}^d \times \mathbb{R}^d.$$

Intuitively, Definition 1 says that the gradient of function f does not change arbitrarily quickly with respect to the parameters. Smoothness assumption is essential for the convergence analysis of the most gradient-based methods. For simplicity, we will use $\|\cdot\|$ to represent the L_2-norm throughout this paper.

Definition 2 (μ-strongly convex) *Function $f : \mathbb{R}^d \to \mathbb{R}$, is μ-strongly convex if there exists a constant $\mu > 0$ such that*

$$f(\bar{x}) \geq f(x) + \nabla f(x)^T(\bar{x} - x) + \frac{\mu}{2}\|\bar{x} - x\|^2, \textit{ for all } (\bar{x}, x) \in \mathbb{R}^d \times \mathbb{R}^d.$$

Definition 3 (convex) *Function $f : \mathbb{R}^d \to \mathbb{R}$, is convex if*

$$f(\bar{x}) \geq f(x) + \nabla f(x)^T(\bar{x} - x), \textit{ for all } (\bar{x}, x) \in \mathbb{R}^d \times \mathbb{R}^d.$$

The convexity assumptions are also essential for the most of convergence analysis in this paper. Note that μ-strongly convexity implies convexity but not vice versa. Throughout the rest of this paper, we assume that under strongly-convex or convex assumptions, there exists an optimal solution x^* such that

$$x^* = \arg\min_{x\in\mathbb{R}^d} f(x).$$

3. Grafting Gradient Descent Algorithm

The key insight of our work is that minibatching and importance sampling can collaborate in a different way. This new technique is called importance resampling which successively employs importance sampling on batch of the subsampled sets. Let $D_m = \{S_m \mid S_m \subset \{1, 2, ..., n\}, |S_m| = m\}$. Technically importance resampling consists of three steps:

- First sample a batch of sets $S_{m_r} \in D_m$. Denote this batch of subsets by $S_m^b = \{S_{m_1}, S_{m_2}, ..., S_{m_b}\}$.

Algorithm 1: Grafting Gradient Descent

Input: The batch size b, subsampled set size m and the learning rate γ.
Initialize: x^0
for $k = 0, 1, 2, ..., T-1$ **do**
Option (a): Sample $S_m^b = \{S_{m_1}, ..., S_{m_b}\}$ with replacement from D_m.
Option (b): Sample $S_m^b = \{S_{m_1}, ..., S_{m_b}\}$ without replacement from D_m.
Compute $P_{S_{m_i}} = \frac{\left(f_{S_{m_i}}(x^k) - f_{S_{m_i},min}\right)}{\sum_{j=1}^{b}\left(f_{S_{m_j}}(x^k) - f_{S_{m_j},min}\right)}$ for $i = 1, 2, ..., b$.
Denote $\mathbf{P} = (P_{S_{m_1}}, ..., P_{S_{m_b}})^\top$.
Resample $\{S_{m_{r_1}}, ..., S_{m_{r_d}}\}$ from S_m^b based on the resampling distribution $\mathbf{P}$.
Compute the grafting gradient as

$$g_{m,b}(x^k) = \begin{pmatrix} \frac{1}{bP_{S_{m_{r_1}}}}\left(\frac{1}{m}\sum_{i\in S_{m_{r_1}}}\frac{\partial f_i}{\partial x_1}(x^k)\right) \\ \vdots \\ \frac{1}{bP_{S_{m_{r_d}}}}\left(\frac{1}{m}\sum_{i\in S_{m_{r_d}}}\frac{\partial f_i}{\partial x_d}(x^k)\right) \end{pmatrix}$$

Update:

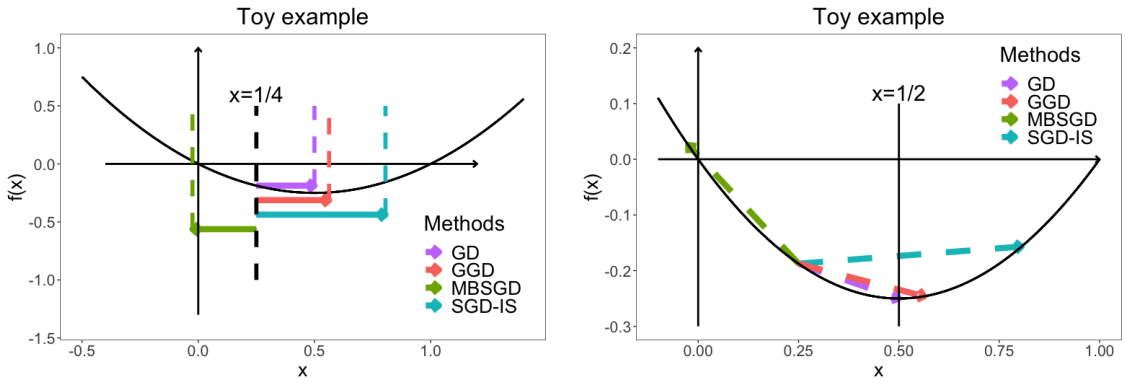
$$x^{k+1} = x^k - \gamma g_{m,b}(x^k)$$

end

- Put a probability measure $\mathbf{P}$ on each element of set S_m^b based on some values.
- Use this probability measure $\mathbf{P}$ to resample d (equals to the dimension of model parameters x) examples from set S_m^b.

The resampling result actually indicates which subset we should use when calculating the partial derivatives in each dimension. As an illustration, assuming that the resampling result is $\{S_{m_{r_1}}, S_{m_{r_2}}, ..., S_{m_{r_d}}\}$ which means that for all $k = 1, 2, ..., d$, we need to calculate the k-th component of the sample average gradient with respect to the subset $S_{m_{r_k}}$, that is, $\partial f_{S_{m_{r_k}}}/\partial x_k$. Combining these total d average partial derivatives, we now construct a grafting gradient which can be used to update the model parameters. The word *grafting* means that this gradient is synthesis and the components of this gradient are determined in a particular way for the purpose of variance reduction. If set S_{m_r} is sampled from D_m without replacement, then the corresponding algorithm is called grafting gradient descent using sampling without replacement (GGD-WoR). Its counterpart is called grafting gradient descent using sampling with replacement (GGD-WR) when S_{m_r} is sampled from D_m independently. The detailed procedure of GGD is shown in Algorithm 1.

From Algorithm 1, intuitively we can get some insights on why GGD may outperform mini-batch SGD and SGD with importance sampling. For the one hand, mini-batch SGD only select one batch of data samples whereas GGD uses important subsets from candidate set S_m^b, which may provide more useful estimations of the full gradient. On the other hand, importance resampling injects additional randomness into GGD compared with SGD with importance sampling, and promotes the diversity of selected data samples, which may profit



(a) (b)

Figure 1: Toy example illustrates the effectiveness of importance resampling

the estimation of the full gradient as well. To get a comprehensive view of GGD method, let us set aside the discussion and focus on a toy example where we have a set of univariate functions $F_s = \{f_1(x), ..., f_{22}(x)\}$ which are defined as

$$f_i(x) = (0.15 + 0.1(i-1))x^2 \text{ , for } i = 1, ..., 20 \text{ and } f_i(x) = -11x \text{ , for } i = 21, 22,$$

and our goal is to minimize the average of these functions which is depicted in Figures 1(a) and 1(b) by black quadratic curves, that is,

$$f(x) = \frac{1}{22}\sum_{i=1}^{22} f_i(x) = x^2 - x.$$

Suppose that the initial starting point is $x = 1/4$ and the learning rate is $\gamma = 1/2$. The best approach that we can take is gradient descent which calculate the full gradient $\nabla f(x)$ at $x = 1/4$ and update x by $x \leftarrow x - \gamma\nabla f(x) = 1/4 + (1/2) * (1/2) = 1/2$ as shown by violet solid line with arrow in Figure 1(a). It is obvious that after one gradient step, we easily find the global minimum of $f(x)$ with the proper learning rate as indicated by the violet dash line in Figure 1(b). Except for gradient descent, we can achieve this goal by other approaches such as mini-batch SGD, mini-batch SGD with important sampling and GGD at a lower computational cost. Now let us dive a little deeper to see how these method will perform in this toy example.

Suppose that for these three methods, we are only allowed to use a size 2 subset of function set F_s. Then for mini-batch SGD, by sampling without replacement, we are likely obtaining a mini batch without $f_{21}(x)$ and $f_{22}(x)$ such like $S_0 = \{f_4(x), f_{17}(x)\}$. Based on that, mini-batch SGD calculate the gradient with respect to S_0 at $x = 1/4$ and update x by $x \leftarrow x - \gamma\nabla f_{S_0}(x) = 1/4 - 11/40 = -1/40$ as shown by green solid line with arrow in Figure 1(a). It moves x in the opposite way of the descent direction and increases the objective function value as indicated by green dash line in Figure 1(b).

For mini-batch SGD with importance sampling, supposing that $f_{21}(x)$ and $f_{22}(x)$ are of great importance so that for a randomly selected index $i_r \in \{1, ..., 22\}$, $P_{21} = P(i_r =$

21) = 0.45, $P_{22} = P(i_r = 22) = 0.45$ and $P_j = P(i_r = j) = 0.005$ for $j = 1, ..., 20$. Based on that, through importance sampling we are likely obtaining $\{f_{21}(x), f_{22}(x)\}$ and as shown by cyan solid line with arrow in Figure 1(a), SGD with importance sampling may update x following

$$x \leftarrow x - \gamma \frac{1}{2}\left(\frac{1}{nP_{21}}\nabla f_{21}(x) + \frac{1}{nP_{22}}\nabla f_{22}(x)\right) = \frac{1}{4} + \frac{11}{22 * 0.45} = \frac{29}{36}.$$

Although SGD with importance sampling moves x along the descent direction, it takes a giant step so that x arrives at 29/36 which is even more far from the optimum point $x = 1/2$ as indicated by cyan dash line in Figure 1(b).

The reason why these two methods fail to minimize the objective function in one step is that sampling without replacement used by mini-batch SGD ignores the valuable information contained by important compenents such as $f_{21}(x)$ and $f_{22}(x)$ which determines the descent direction, and importance sampling lacks the randomness to some extent so that the selected components are individually very informative but not necessarily so jointly. Using importance sampling may form a batch of important but redundant components which lacks diversity and may negatively impact the performance of mini-batch SGD with importance sampling.

For GGD, if we implement importance resampling which only resamples one subset out of $b = 10$ randomly selected subsets $\{S_{m_1}, ..., S_{m_{10}}\}$ in our toy example, then 10 randomly selected subsets may contain only one subset such like $S_{m_3} = \{f_{11}(x), f_{21}(x)\}$ with important component $f_{21}(x)$ since the probability of 10 randomly selected subsets containing at least one important component is

$$P\left(\exists S_{m_j} \in \{S_{m_1}, ..., S_{m_{10}}\} \text{ s.t. } f_{21}(x) \in S_{m_j} \text{ or } f_{22}(x) \in S_{m_j}\right) = 1 - \left(\frac{\mathrm{C}_{20}^2}{\mathrm{C}_{22}^2}\right)^{10} \approx 0.8582,$$

where $\mathrm{C}_n^k = n!/(k!(n-k)!)$ is the number of k-combinations from a given set of n elements. Then through the resampling procedure, this subset $\{f_{11}(x), f_{21}(x)\}$ is likely to be the chosen one which is used to calculate the grafting gradient, and as shown by red solid line with arrow in Figure 1(a) GGD may update x following

$$x \leftarrow x - \gamma \frac{1}{10P_{S_{m_3}}}\left(\frac{1}{2}\nabla f_{11}(x) + \frac{1}{2}\nabla f_{21}(x)\right) \approx 0.5629.$$

Although GGD fails to find the global minimizer in one gradient step, it moves x quite near to the optimum point and decreases the objective function value most compared with mini-batch SGD and SGD with important sampling as indicated by red dash line in Figure 1(b). Importance resampling can guarantee us a subset which contains important and diverse components with high probability, and consequently GGD can provide more informative and accurate estimation of the full gradient. The relationship among mini-batch SGD, SGD with importance sampling and GGD reminds us of the notion of exploration vs exploitation in many other domains. From this perspective, GGD can be regarded as the trade off between the "exploitation" and "exploration", and we hope that GGD could achieve comparable results when applying to more complex and high-dimensional problems.

Now let us get back to technical work and put some remarks on the resampling distribution. In the GGD method, the noise variance comes from $\mathbb{E}\|g_{m,b}(x^k)\|^2$ which is equivelant to

$$\begin{aligned}\mathbb{E}\|g_{m,b}(x^k)\|^2 &= \mathbb{E}\left[\mathbb{E}\left[\frac{1}{b^2}\sum_{j=1}^{d}\left(\frac{1}{P_{S_{m_{r_j}}}}\right)^2\left(\frac{\partial f_{S_{m_{r_j}}}}{\partial x_j}\right)^2 \mid S_m^b\right]\right] \\ &= \mathbb{E}\left[\frac{1}{b^2}\sum_{j=1}^{d}\sum_{i=1}^{b}\frac{1}{P_{S_{m_i}}}\left(\frac{\partial f_{S_{m_i}}}{\partial x_j}\right)^2\right] \\ &= \mathbb{E}\left[\frac{1}{b^2}\sum_{i=1}^{b}\frac{1}{P_{S_{m_i}}}\|\nabla f_{S_{m_i}}(x^k)\|^2\right]\end{aligned}$$

Given x^k and $S_m^b = \{S_{m_1}, ..., S_{m_b}\}$, denoting that Δ^b is the b-dimensional simplex, we know that the solution of

$$\min_{\mathbf{P}\in\Delta^b}\sum_{i=1}^{b}\frac{1}{P_{S_{m_i}}}\|\nabla f_{S_{m_i}}(x^k)\|^2 \tag{1}$$

is

$$P_{Sm_i}^{opt} = \frac{\|\nabla f_{S_{m_i}}(x^k)\|}{\sum_{j=1}^{b}\|\nabla f_{S_{m_j}}(x^k)\|}, \quad i = 1, 2, ..., b. \tag{2}$$

The optimal resampling distribution defined in (2) minimizes the noise variance of grafting gradient. However, deriving this optimal resampling distribution requires evaluations of bmd partial derivatives. In analogy with Zhao and Zhang (2015), if we assume that the individual loss function f_i is L_i-smooth, bounded below by $f_{i,min}$, then $\|\nabla f_{S_{m_i}}(x^k)\|^2$ can be bounded by

$$\|\nabla f_{S_{m_i}}(x^k)\|^2 \le 2L_{S_{m_i}}\left(f_{S_{m_i}}(x^k) - f_{S_{m_i},min}\right) \le 2L_{max}\left(f_{S_{m_i}}(x^k) - f_{S_{m_i},min}\right),$$

where $L_{max} = \max_{i\in[n]}\{L_i\}$. $f_{S_{m_i},min}$ can be easily estimated if we know a uniform lower bound for $f_i(x^k)$ and fortunately if loss function f_i is non-negative, then we can let $f_{S_{m_i},min} = 0$ although it may rather be pessimistic. Noting that $\|\nabla f_{S_{m_i}}(x^k)\|^2$ can be bounded by $2L_{max}\left(f_{S_{m_i}}(x^k) - f_{S_{m_i},min}\right)$, then we can relax the optimization problem (1) following the analysis provided by Zhao and Zhang (2015) as

$$\min_{\mathbf{P}\in\Delta^b}\sum_{i=1}^{b}\frac{1}{P_{S_{m_i}}}\|\nabla f_{S_{m_i}}(x^k)\|^2 \le \min_{\mathbf{P}\in\Delta^b}\sum_{i=1}^{b}\frac{2L_{max}}{P_{S_{m_i}}}\left(f_{S_{m_i}}(x^k) - f_{S_{m_i},min}\right).$$

Since L_{max} is independent of timestep and subset selection, the optimal resampling probability can be approximated by $P'_{S_{m_i}} \propto \left(f_{S_{m_i}}(x^k) - f_{S_{m_i},min}\right)^{1/2}$. To simplify our analysis, we use a resampling probability in "square" form instead, that is,

$$P_{S_{m_i}} = \frac{\left(f_{S_{m_i}}(x^k) - f_{S_{m_i},min}\right)}{\sum_{j=1}^{b}\left(f_{S_{m_j}}(x^k) - f_{S_{m_j},min}\right)}. \tag{3}$$

It is clear that in one iteration, Algorithm 1 only requires evaluations of mb loss function values and md partial derivatives which is computationally cheaper than deriving the optimal resampling probability. We do not adopt the one-shot resampling probability such as Zhao and Zhang (2015) where $P_{S_{m_i}} \propto L_{S_{m_i}}$ because one drawback of the one-shot resampling probability, which is defined in terms of the L-smoothness constants, is that it may not be applicable in neural network setting where the individual L-smoothness constant does not has a closed form and requires additional techniques to estimate. On the contrary, since the individual loss function value can be explicitly obtained through the forward-propagtion of neural network, the resampling probability $P_{S_{m_i}}$ redresses the flaw of one-shot resampling probability albeit at a higher computational cost.

For numerical stability, without loss of generality, we assume that $P_{S_{m_i}} > 0$ for $i = 1, ..., b$. If there exist some $S_{m_p} \in \{S_{m_1}, ..., S_{m_b}\}$ such that $f_{S_{m_p}}(x^k) - f_{S_{m_p},min} = 0$, then the corresponding sample will not show up in the grafting gradient. When such a situation happens, we should remove this sample and add a new one until we put non-zero mass on every element in the set S_m^b. Furthermore, if there is no set S_m^b which satisfies this constraint, we just put equal mass on every sample and resample them to construct the grafting gradient. We also notice that the components of the grafting gradient in Algorithm 1 are reweighted by the batch size and corresponding probabilities to ensure the unbiasedness of grafting gradient with respect to the full gradient.

4. Convergence Results for GGD

Now we can present the convergence results for Algorithm 1 under different assumptions. Convergence results for GGD-WR and GGD-WoR are presented in tandem.

4.1 Convergence Result for GGD-WR under Strongly-convex Assumption

Under the assumptions of strong convexity and L-smoothness, a theoretical bound can be obtained as follows.

Theorem 4 *Suppose that the objective function f is L-smooth, the individual loss function f_i is μ-strongly convex, L_i-smooth, bounded below by $f_{i,min}$ for all $i \in [n]$. When Algorithm 1 is run with the fixed stepsize where $\gamma < \min\{2/\mu, 2b/\left(C + 2\bar{L}(b-1)\right)\}$ and option (a), the iterates generated by Algorithm 1 satisfy*

$$\mathbb{E}\left[\left\|x^T - x^*\right\|^2\right] \leq (1-\mu\gamma)^T \left\|x^0 - x^*\right\|^2 + \frac{2\gamma R}{\mu b D},$$

where $\bar{L} = (1/n)\sum_{i=1}^n L_i$, $R = f(x^*) - f_{min}$, $f_{min} = (1/n)\sum_{i=1}^n f_{i,min}$,

$$C = \left(\frac{2L_{max}(n-m) + 2n(m-1)L}{m(n-1)}\right) \text{ and } D = \left(\frac{L_{max}(n-m)}{m(n-1)} + (b-1)\bar{L}\right)^{-1},$$

are constants which are independent of iteration number T.

From Theorem 4, we know that the iterates generated by GGD-WR with the fixed stepsize converge at a linear rate up to some noise level. Let us take a deep look at the noise level $2\gamma R/\mu b D$. It seems that some constants may explode such as C and D since they

show dependence on the size of training set n, but actually not for any reasonable values of m and b. For fixed n, C is monotonically decreasing with respect to the subsampled set size m, thus C can be bounded by $2L_{max}$. As for D, since $(n-m)/m$ is decreasing with respect to m, then $1/bD$ can be bounded by $L_{max} + \bar{L}$, which proves that $2\gamma R/\mu bD$ does not explode for any possible values of m and b. To see the superiority and robustness of GGD-WR method, we derive the theoretical bounds for vanilla SGD, mini-batch SGD and SGD with importance sampling under the same assumptions. For the iterates generated by SGD, they satisfy

$$\mathbb{E}\|x_{SGD}^T - x^*\|^2 \leq (1-\mu\gamma)^T \mathbb{E}\|x^0 - x^*\|^2 + \frac{2\gamma L_{max} R}{\mu}. \tag{4}$$

The derivation for equation (4) can be found in Appendix A.2. After some straightforward calculations, we know that $2R/bD \leq 2L_{max}R$ holds for any b and $m \in \mathbb{N}$, which implies that by importance resampling, variance brought by the grafting gradient is always less than the variance brought by the stochastic gradient in every iteration. Hence the training process can be boosted when replacing stochastic gradient with grafting gradient in each step. Although GGD-WR has the same convergence rate as SGD, it reduces the noise variance to some extent. In other words, the solution path found by the GGD-WR algorithm fluctuates in a smaller neighborhood of the optimum value compared with the vanilla SGD method. Batch size b and the size of subsampled sets m influences the radius of this neighborhood. The larger m is, the smaller the radius is. The effect brought by b depends on the relationship between $L_{max}(n-m)/m(n-1)$ and $\bar{L}$. That is, if $\bar{L} \geq L_{max}(n-m)/m(n-1)$, then increasing b will enlarge the radius of this neighborhood, otherwise, increasing b leads to narrowing the radius of this neighborhood. We also derive the bounds for mini-batch SGD and SGD with importance sampling. For the latter one, the sampling probability is given by $P_i = L_i / \sum_{j=1}^n L_j$ for all $i \in [n]$. Under the same assumptions, their convergence bounds are given as follows.

- For mini-batch SGD, the iterates satisfy

$$\mathbb{E}\|x_{mSGD}^T - x^*\|^2 \leq (1-\mu\gamma)^T \mathbb{E}\|x^0 - x^*\|^2 + \frac{2\gamma R L_{max}(n-m)}{m(n-1)\mu}. \tag{5}$$

- For SGD with importance sampling, the iterates satisfy

$$\mathbb{E}\|x_{ISSGD}^T - x^*\|^2 \leq (1-\mu\gamma)^T \mathbb{E}\|x^0 - x^*\|^2 + \frac{2\gamma \bar{L} R}{\mu}. \tag{6}$$

From (5), (6) and Theorem 4, we can see that the noise variance variance of GGD-WR is the weighted sum of the noise variance of mini-batch SGD and the noise variance of SGD with importance sampling with weights $(1/b, (b-1)/b)$. This result implies two properties of GGD-WR.

- **Robustness:** $2R/bD$ is not greater than $\max\{2L_{max}R(n-m)/m(n-1), 2\bar{L}R\}$.
- **Tendency:** Batch size b controls the weights. The larger b is, the closer the noise variance of GGD approaches to the noise variance of SGD with importance sampling.

For the former one, we say that grafting gradient is a *doubly robust* estimator with respect to the full gradient in sense that its theoretical bound is superior to the worse one of mini-batch SGD and SGD with importance sampling. Doubly robust property may be of great help when we do not know whether mini-batch SGD or SGD with importance sampling will surpass for real problems. In section 6, we empirically compare these four method (vanilla SGD, GGD, mini-batch SGD and SGD with importance sampling) and bring out more discussions about their superiorities and applicabilities. For the latter one, Although in GGD-WR algorithm b can be arbitrarily large so that the tremendous computational cost brought by loss function evaluation may be unaffordable, the tendency property of GGD suggests that GGD does not benefit and even can be harmed from blindly increasing batch size b when the noise variance of mini-batch SGD is smaller than the noise variance of SGD with importance sampling. Thus for the rest of our paper, unless specifying, batch size b is assigned by default a relatively small value such like $b \in \{2, 3, 4\}$ which balances the trade off between the computational cost and the performance, and moreover implies that the computational cost of loss function evaluations is insignificant compared with the computational cost of partial derivatives evaluations.

The convergence analysis would be incomplete without considering how theoretical results impact on the computational workload when the stochastic optimization methods are applied for the real problems. *Complexity* results give the bound for the total number of partial derivatives evaluations as the main computational complexity to achieve the ϵ-optimality in expectation. The ϵ-optimality in expectation is defined as a point x satisfies

$$\mathbb{E}\|\nabla f(x)\|^2 \leq \epsilon, \text{ or } \mathbb{E}\|x - x^*\|^2 \leq \epsilon, \text{ or } \mathbb{E}\left[f(x) - f(x^*)\right] \leq \epsilon,$$

where x^* is assumed to be the global minimizer of f. Unless specifying, convergence and complexity results will be provided in tandem for the rest of this paper.

Corollary 5 *If we choose stepsize* $\gamma = \min\{2/\mu, 2b/\left(C + 2\bar{L}(b-1)\right), \epsilon\mu bD/4R\}$, *then to achieve* ϵ*-optimality, the total iteration number* T *should satisfy*

$$T \geq \max\left\{\frac{1}{2}, \frac{(C + (b-1)2\bar{L})}{2b\mu}, \frac{4R}{\epsilon\mu^2 bD}\right\} \ln\left(\frac{2\mathbb{E}\|x^0 - x^*\|^2}{\epsilon}\right).$$

Hence the total complexity to achieve ϵ*-optimality is*

$$md \cdot \max\left\{\frac{1}{2}, \frac{(C + (b-1)2\bar{L})}{2b\mu}, \frac{4R}{\epsilon\mu^2 bD}\right\} \ln\left(\frac{2\mathbb{E}\|x^0 - x^*\|^2}{\epsilon}\right).$$

Suppose that $m \ll n$, then the iteration complexity result will be $\mathcal{O}\left(d/\epsilon \ln(1/\epsilon)\right)$. Combining this result with Theorem 4, we know that compared with vanilla SGD method, the noise variance of GGD-WR method shrinks by a constant factor so that GGD-WR improves a non-dominant term in complexity. A diminishing stepsize sequence is another choice to reduce the noise variance. Its convergence result is shown as follows.

Theorem 6 *Suppose that the objective function* f *is* L*-smooth, the individual loss function* f_i *is* μ*-strongly convex and* L_i*-smooth, bounded below by* $f_{i,min}$ *for all* $i \in [n]$. *When*

Algorithm 1 is run with option (a) and a stepsize sequence which satisfies for all $k = 0, 1, 2, ...,$

$$\gamma_k = \frac{p}{q+k} \quad \textit{for some} \quad p > \frac{1}{\mu} \quad \textit{and} \quad q > 0 \quad \textit{such that} \quad \frac{p}{q} < \frac{2b}{(C + 2(b-1)\bar{L})}.$$

Then for each k*, the expected optimality gap satisfies*

$$\mathbb{E}\|x^k - x^*\|^2 \leq \frac{v}{q+k},$$

where $v = \max\{2p^2R/(p\mu - 1)bD, p\mathbb{E}\left[\|x^0 - x^*\|^2\right]\}$.

Since $1/bD$ can be bounded for any possible values of m and b, v will not explode and the expected optimality gap can be controlled with large enough iteration number T. Theorem 6 shows that when using a diminishing stepsize sequence, the iterates generated by Algorithm 1 with option (a) can converge to the optimum point x^* at a sublinear rate. Although the convergence rate is much slower than we have obtained in Theorem 4, it eliminates the variance brought by grafting gradient and reduces the training time. Theorem 6 also indicates that the total iteration number T should satisfy $T \geq (v/\epsilon) - q$ to achieve ϵ-optimality.

Corollary 7 $\forall \epsilon > 0$*, the total complexity to achieve* ϵ*-optimality is*

$$md \cdot \left(\frac{v}{\epsilon} - q\right)$$

Again if $m \ll n$, then the complexity result will be $\mathcal{O}(d/\epsilon)$ which indicates that using the grafting gradient only improves a non-dominant term in complexity. Although the convergence rate is sublinear, the complexity bound improves from $\mathcal{O}\left((d/\epsilon)\ln 1/\epsilon\right)$ to $\mathcal{O}\left(d/\epsilon\right)$, which means that GGD-WR with the diminishing stepsize sequence requires less iterations to achieve the same level ϵ-optimality. The cause behind this counterintuitive phenomenon is that to achieve ϵ-optimality, the fixed stepsize should keep the same order as ϵ, while for the diminishing stepsize sequence, it begins with p/q and gradually decreases.

4.2 Convergence Result for GGD-WR under Convex Assumption

In this section, we study the convergence property of the GGD-WR method for the general convex individual loss function. For simplicity, we do not compare the theoretical bounds of GGD-WR, SGD, mini-batch SGD and SGD with importance sampling under convex or non-convex assumptions.

Theorem 8 *Suppose that the objective function* f *is* L*-smooth, the individual loss function* f_i *is* L_i*-smooth and convex, bounded below by* $f_{i,min}$ *for all* $i \in [n]$*. When Algorithm 1 is run with* $\gamma < 3b/2\left(C + 2\bar{L}(b-1)\right)$ *and option (a), denoting* $\hat{x} = \frac{1}{T}\sum_{k=0}^{T-1} x^k$*, then it satisfies*

$$\mathbb{E}\left[f(\hat{x}) - f(x^*)\right] \leq \frac{2\mathbb{E}\|x^0 - x^*\|^2}{T\gamma} + \frac{4\gamma R}{bD}. \tag{7}$$

We notice that when strongly-convex assumption does not hold, the expected difference in objective function value between any iterate and the optimum point x^*, $\mathbb{E}\left[f(x^k) - f(x^*)\right]$, does not have a quadratic lower bound, that is, can not be bounded below by $\mathcal{O}\left(\mathbb{E}\|x^k - x^*\|^2\right)$. The non-existence of a quadratic lower bound influences the convergence property of GGD-WR method. As derived in Theorem 8, when a fixed stepsize is used, the GGD-WR method only converges at a sublinear rate up to some noise level for general convex objective function. Consequently, we have the complexity result in the general convex case.

Corollary 9 *If we take the stepsize which satisfies*

$$\gamma = \min\{3b/2(C + (b-1)2\bar{L}), \epsilon bD/8R\},$$

to achieve ϵ-optimality, then total iteration number T should satisfy

$$T \geq \frac{4\mathbb{E}\|x^0 - x^*\|^2}{\epsilon \min\{3b/2(C + (b-1)2\bar{L}), \epsilon bD/8R\}}.$$

Thus the total complexity is

$$md \cdot \frac{4\mathbb{E}\|x^0 - x^*\|^2}{\epsilon \min\{3b/2(C + (b-1)2\bar{L}), \epsilon bD/8R\}}.$$

From Corollary 5 and 9, we see how the convergence results impact on the complexity results. Since the GGD method does not possess a linear convergence rate under general convex assumption, for $m \ll n$, the complexity is $\mathcal{O}(d/\epsilon^2)$ which is far larger than $\mathcal{O}(d/\epsilon)$.

4.3 Convergence Result for GGD-WR under Non-convex Assumption

Following the analysis provided by Khaled and Richtárik (2020), we can take a step towards the theoretical bound for GGD-WR method without any additional assumption at all and provide the following convergence results.

Theorem 10 *Suppose that objective function f is L-smooth, the individual loss function f_i is L_i-smooth, bounded below by $f_{i,min}$ for all $i \in [n]$. When Algorithm 1 is run with option (a) and fixed stepsize $\gamma \leq (2b-1)/L$, then the iterates generated by Algorithm 1 satisfy*

$$\min_{k=0,1,\ldots,T-1} \mathbb{E}\|\nabla f(x^k)\|^2 \leq \frac{2\gamma L}{D}\left(1 + \frac{bD}{\gamma^2 LT}\right)\delta_0,$$

where $\delta_0 = \mathbb{E}\left[f(x^0)\right] - f_{min}$ is a constant.

Non-convex GGD-WR shows a sublinear convergence rate up to some noise level as well. Difference between convex GGD-WR and non-convex GGD-WR is that the theoretical bound of non-convex GGD-WR shows possible divergence since $1/D$ is monotonically increasing with respect to b. However, the minimum of expected gradient norm can be controlled arbitrarily small by manually assigning b a small value and using a designated stepsize. This bound is in fact optimal for GGD-WR without additional assumptions on second-order smoothness such like Polyak-Łojasiewicz condition since the convergence rate of non-convex GGD-WR can not exceed the convergence rate of convex GGD-WR. The following corollary states the complexity results to attain the ϵ-stationary point.

Corollary 11 *With the stepsize*

$$\gamma = \min\left\{\frac{2b-1}{L}, \frac{\epsilon D}{4L\delta_0}\right\},$$

Algorithm 1 with option (a) can still achieve ϵ-optimality as long as the iteration number T satisfies

$$T \geq \frac{4\delta_0 b}{\epsilon}\max\left\{\frac{L}{2b-1}, \frac{4L\delta_0}{\epsilon D}\right\}.$$

Hence to achieve ϵ-optimality, the total complexity is

$$md \cdot \frac{4\delta_0 b}{\epsilon}\max\left\{\frac{L}{2b-1}, \frac{4L\delta_0}{\epsilon D}\right\}.$$

When the strongly convex assumption does not hold, for $m \ll n$, the complexity bound degrades from $\mathcal{O}(d/\epsilon)$ to $\mathcal{O}(d/\epsilon^2)$. In other words, it takes more time to achieve the same level ϵ-optimality for general convex or non-convex functions. Although the complexity bounds of convex GGD-WR and non-convex GGD-WR are identical for $m \ll n$, the average objective function value can be bounded under convex assumption, while we can only control the minimum expected gradient norm for non-convex objective function.

4.4 Convergence Result for GGD-WoR under Strongly-convex Assumption

In the rest of Section 4, we mainly focus on the convergence results of GGD-WoR under different assumptions. Through the analysis of GGD-WoR, we find that while the doubly robust property does not hold anymore, GGD-WoR can be considered as a more general framework since it can include vanilla SGD, mini-batch SGD and SGD with importance sampling as special cases. We first derive the unified theoretical bound under strongly-convex assumption and show how GGD-WoR connects those classic stochastic optimization methods.

Theorem 12 *Suppose that the objective function f is L-smooth, the individual loss function f_i is μ-strongly convex, L_i-smooth, bounded below by $f_{i,min}$ for all $i \in [n]$. When Algorithm 1 is run with the fixed stepsize where $\gamma \leq \min\left\{2/\mu, b^2m^2/M\right\}$ and option (b), the iterates generated by Algorithm 1 satisfy*

$$\mathbb{E}\left[\left\|x^T - x^*\right\|^2\right] \leq (1-\mu\gamma)^T\left\|x^0 - x^*\right\|^2 + \frac{2\gamma RM}{\mu b^2 m^2}, \tag{8}$$

where $M = \left(n^2 \cdot M_2 \cdot \bar{L} + n(M_1 - M_2)\cdot \tilde{L}\right)$,

$$M_1 = \frac{mb}{n} + \frac{mb(\mathrm{C}_{n-1}^{m-1}-1)(b-1)}{n(\mathrm{C}_n^m - 1)},\ M_2 = \frac{mb(b-1)\mathrm{C}_{n-1}^{m-1}}{n(\mathrm{C}_n^m-1)} + \frac{mb(m-1)(\mathrm{C}_n^m - b)}{n(n-1)(\mathrm{C}_n^m-1)},$$

$\tilde{L} = \mathbb{I}_{\{M_1 \geq M_2\}}\cdot L_{max} + \mathbb{I}_{\{M_1 < M_2\}}\cdot L_{min}$ *and* $L_{min} = \min_{i\in[n]} L_i$ *are constants which are independent of iteration number T.*

Theorem 12 suggests that the iterates generated by GGD-WoR converge at a linear rate up to some noise level similar to that of GGD-WR algorithm. To see whether M/b^2m^2 will explode for any reasonable values of m, b and n, we first study two quantities M_1 and M_2. $M_1 = \mathcal{O}(m^2b^2/n^2)$ and $M_2 = \mathcal{O}(m^2b^2/n^2)$ implies that $M/b^2m^2 = \mathcal{O}\left(1 + 1/n\right)$, thus the noise variance does not explode under any circumstances. It is also clear that since the noise variance of GGD-WoR is related to the minimum of the individual smoothness constants, doubly robust property does not hold for an arbitrary choice of $m \in \mathcal{N}^+$, $b \in [\mathrm{C}_n^m]$. Luckily, we find that for certain configurations of b and m, the procedures of GGD-WoR are identical to that of vanilla SGD, mini-batch SGD and the theoretical bound of GGD-WoR is identical to that of SGD with importance sampling in expectation. We first verify the case for vanilla SGD.

Proposition 13 *If $m = 1$ and $b = 1$, then the bound given by Theorem 12 is equivalent to (4).*

When $m = 1$ and $b = 1$, intuitively, this configuration indicates that there is no resampling at all and only one stochastic sampled gradient can be used, which is exactly how vanilla SGD works in practice. By some straightforward calculations, we can also obtain that $M_1 = 1/n$ and $M_2 = 0$, then the theoretical upper bound for the noise variance of GGD-WoR is $2\gamma RL_{max}/\mu$, the same as the result derived in (4).

Proposition 14 *If $m \in \mathcal{N}^+$ and $b = 1$, then the bound given by Theorem 12 is an upper bound of (5).*

We use $\mathcal{N}^+$ to represent the positive natural numbers. Likewise, for any $m \in \mathcal{N}^+$ and $b = 1$, the procedure of GGD-WoR algorithm is identical to the procedure of mini-batch SGD with size m. As for the theoretical bounds, they are not identical since if we substitute $M_1 = m/n$, $M_2 = m(m-1)/n(n-1)$ into (8), then we have

$$\mathbb{E}\left[\left\|x^T - x^*\right\|^2\right] \leq (1-\mu\gamma)^T \left\|x^0 - x^*\right\|^2 + \frac{2\gamma R}{\mu m^2}\left(\frac{nm(m-1)}{n-1}\bar{L} + \frac{m(n-m)}{n-1}L_{max}\right) \quad (9)$$

Apparently, the right side of (9) is an upper bound for the right side of (5). From Lemma 32 in Appendix, we know that if we further bound $\|\nabla f(x^k)\|^2$ by $2\bar{L}\left(f(x^k) - f_{min}\right)$, then we can derive the same bound as (9). Under this configuration, (8) gives the sub-optimal upper bound for the expected optimality gap, this bound can be greatly improved by some technical tricks as provided by proof of Theorem 4 in Appendix.

Proposition 15 *If $m = 1$ and $b = n$, then the bound given by Theorem 12 is equivalent to (6).*

Although the sampling probability given in Algorithm 1 and even the procedure of GGD-WoR are different from that of SGD with importance sampling, the theoretical bound given by (8) and (6) are identical since $M_1 = M_2 = 1$. In that sense, GGD-WoR with $m = 1$ and $b = n$ is identical to SGD with importance sampling in expectation. These three special cases prove that GGD-WoR can be regarded as a more general stochastic optimization framework. The following statement gives the complexity result for GGD-WoR under strongly-convex assumption.

Corollary 16 *If we choose stepsize*

$$\gamma = \min\left\{\frac{2}{\mu}, \frac{b^2m^2}{M}, \frac{\epsilon\mu b^2m^2}{4RM}\right\},$$

then to achieve ϵ-optimality, the total iteration number T should satisfy

$$T \geq \max\left\{\frac{1}{2}, \frac{M}{b^2m^2\mu}, \frac{4RM}{\epsilon\mu^2b^2m^2}\right\}\ln\left(\frac{2\mathbb{E}\|x^0 - x^*\|^2}{\epsilon}\right).$$

Hence the total complexity to achieve ϵ-optimality is

$$md \cdot \max\left\{\frac{1}{2}, \frac{M}{b^2m^2\mu}, \frac{4RM}{\epsilon\mu^2b^2m^2}\right\}\ln\left(\frac{2\mathbb{E}\|x^0 - x^*\|^2}{\epsilon}\right).$$

The unified complexity result also include vanilla SGD, mini-batch SGD and SGD with importance sampling as special cases. For example, if $m = 1$, $b = 1$, then the complexity result will be $\mathcal{O}\left((d/\epsilon)\ln(1/\epsilon)\right)$ which is identical to that of vanilla SGD in other literatures (Gower et al., 2019). It is worth noting that although Corollary 16 give the right complexity of SGD with importance sampling, it does not give the proper complexity for GGD-WoR itself. In contrast with one-shot sampling probability given by Zhao and Zhang (2015), the resampling probability $P_{S_{m_i}} \propto \left(f_{S_{m_i}}(x^k) - f_{S_{m_i},min}\right)$, which is iterate-dependent, needs to be calculated in every iteration. Hence when $m = 1$, $b = n$, the computational cost for evaluating n loss function values in one iteration can not be ignored and thus for extremely large b, the complexity results of GGD should be the bound for partial derivatives and loss function evaluations. Under this definition, the complexity results of GGD with $b = n$ and $m = 1$ should be

$$(n + d) \cdot \max\left\{\frac{1}{2}, \frac{\bar{L}}{\mu}, \frac{4R\bar{L}}{\epsilon\mu^2}\right\}\ln\left(\frac{2\mathbb{E}\|x^0 - x^*\|^2}{\epsilon}\right),$$

which is far larger than the bound obtained by Corollary 16. In analogy with GGD-WR, the noise variance can be further reduced by a diminishing stepsize sequence. For simplicity, we do not present the convergence and complexity results for GGD-WoR using a diminishing stepsize sequence. Similar result can be obtained following the analysis provided in Theorem 6.

4.5 Convergence Result for GGD-WoR under Convex Assumption

In analogy with Theorem 12, we can obtain a unified theoretical bound for GGD-WoR under convex assumption.

Theorem 17 *Suppose that the objective function f is L-smooth, the individual loss function f_i is L_i-smooth and convex, bounded below by $f_{i,min}$ for all $i \in [n]$. When Algorithm 1 is run with $\gamma < b^2m^2/2M$ and option (b), denoting $\hat{x} = \frac{1}{T}\sum_{k=0}^{T-1} x^k$, then it satisfies*

$$\mathbb{E}\left[f(\hat{x}) - f(x^*)\right] \leq \frac{\mathbb{E}\|x^0 - x^*\|^2}{T\gamma} + \frac{2\gamma RM}{b^2m^2}.$$

Theorem 17 shows that under convex assumption, GGD-WoR can converge at a sublinear rate up to some noise level. Likewise from Theorem 17, we can also obtain the convergence resuls for vanilla SGD, mini-batch SGD and SGD with importance sampling under convex assumption. For simplicity, we do not illustrate these results in detail. The corresponding complexity result for GGD-WoR method is given as follows.

Corollary 18 *If we take the stepsize which satisfies*

$$\gamma = \frac{b^2m^2}{2M}\min\left\{1, \frac{\epsilon}{2R}\right\},$$

to achieve ϵ-optimality, then the total iteration number T should satisfy

$$T \geq \frac{4M\mathbb{E}\|x^0 - x^*\|^2}{\epsilon b^2m^2\min\{1, \epsilon/2R\}}.$$

Thus the total complexity is

$$d \cdot \frac{4M\mathbb{E}\|x^0 - x^*\|^2}{\epsilon b^2m\min\{1, \epsilon/2R\}}.$$

Corollary 18 shows that for $m \ll n$, the complexity result for GGD-WoR will be $\mathcal{O}(d/\epsilon^2)$ which is same as that of vanilla SGD, mini-batch SGD with same mini batch size m and SGD with importance sampling since these improved methods all shrink the noise variance of the stochastic sampled gradient by some constant factors and only improve some non-dominant terms in complexity.

4.6 Convergence Result for GGD-WoR under Non-convex Assumption

Finally, we can formally give the convergence result for GGD-WoR under non-convex assumption.

Theorem 19 *Suppose that objective function f is L-smooth, the individual loss function f_i is L_i-smooth, bounded below by $f_{i,min}$ for all $i \in [n]$. When Algorithm 1 is run with option (b) and fixed stepsize $\gamma \leq \epsilon b^2m^2/2LM\delta_0$ where $\epsilon > 0$, then the iterates generated by Algorithm 1 satisfy*

$$\min_{k=0,1,\ldots,T-1}\mathbb{E}\|\nabla f(x^k)\|^2 \leq \frac{\epsilon}{2} + \frac{\delta_0}{\gamma T},$$

where $\delta_0 = \mathbb{E}\left[f(x^0)\right] - f_{min}$ is a constant.

Theorem 19 shows that with a hand-picked stepsize, GGD-WoR can still converge at a sublinear rate up to an arbitrary small noise level, which indicates that the minimum expected gradient norm can be effectively controlled by increasing iteration number T. The complexity results for GGD-WoR can be directly derived from the above theoretical bound.

Corollary 20 *With the stepsize*

$$\gamma = \frac{\epsilon b^2m^2}{2LM\delta_0},$$

Algorithm 1 with option (b) can still achieve ϵ-optimality as long as the iteration number T satisfies

$$T \geq \frac{4LM\delta_0^2}{\epsilon^2 b^2 m^2}.$$

Hence to achieve ϵ-optimality, the total complexity is

$$d \cdot \frac{4LM\delta_0^2}{\epsilon^2 b^2 m}.$$

For $m \ll n$, the complexity result of GGD-WoR will be $\mathcal{O}(d/\epsilon^2)$, which indicates that it is much slower for GGD-WoR method to reach the ϵ-optimality without convexity and any extra asuumption on smoothness. Combining all the theorems and corollaries provided in Section 4, we find that although GGD-WR and GGD-WoR have idiosyncratic noise levels with the fixed stepsize, their convergence rates and corresponding complexity results are identical under same assumptions: $\mathcal{O}((d/\epsilon)\ln(1/\epsilon))$ for strongly-convex objective function and $\mathcal{O}(d/\epsilon^2)$ for convex or non-convex objective function. These results suggest that the performances of GGD-WR and GGD-WoR methods may be quite indistinguishable for identical m and b when solving the real problems.

5. Variance Reduction Method and Adaptive Stepsize Method

In Section 4, we know that to achieve ϵ-optimality, GGD-WR and GGD-WoR methods rely on a small stepsize or diminishing stepsize sequence which results in a pretty slow convergence in practice. If we insist on using the fixed stepsize, then a better technique that reduces the variance with a fixed stepsize is required. Recalling that based on the SGD framework, lots of advanced variance reduction methods like SVRG can significantly improve the performance compared with the vanilla SGD method. In the first part of this section, we give one example to show the compatibility of variance reduction method and grafting gradient, and illustrate the convergence results of this composite method. If the fixed stepsize is abnegated, another way to accelerate the training process is through adaptive stepsize methods like Adam. We hybridize GGD-WR with Adam, propose GGD-WR-Adam method and provide its convergence results with additional assumptions in the other part of this section.

5.1 GGD-WR-SVRG Method

The key idea of SVRG adopts from a variance reduction technique which is commonly used in sampling theory called control variates. In the SVRG method, control variates is used to modify the stochastic sample gradient so that SVRG method shows a linear convergence rate. Since the grafting gradient can play the same role as stochastic gradient, we can bring out a modified grafting gradient to update the parameters as well. The proposed GGD-WR-SVRG is shown in Algorithm 2. Option (a) was first proposed in Free-SVRG (Sebbouh et al., 2019) and p_k, $k \in \{0, ..., q-1\}$ are defined as follows.

$$V_q = \sum_{k=0}^{q-1}(1-\gamma\mu)^{q-1-k} \text{ and } p_k = \frac{(1-\gamma\mu)^{q-1-k}}{V_q}, \text{ for } k = 0, ..., q-1, \tag{10}$$

Algorithm 2: GGD-WR-SVRG method

Input: Batch size b, subsampled set size m, learning rate γ and update period q.
Initialize: x_0^q and set $\bar{x}_0 = x_0^q$, $x_1^0 = x_0^q$
for $s = 1, 2, ..., T$ **do**
  $\bar{x} = \bar{x}_{s-1}$
  Compute $\bar{\mu} = \nabla f(\bar{x})$
  for $k = 0, ..., q-1$ **do**
    Sample $S_m^b = \{S_{m_1}, ..., S_{m_b}\}$ with replacement from D_m
    Compute $P_{S_{m_i}} = \frac{\|\nabla f_{S_{m_i}}(x_s^k) - \nabla f_{S_{m_i}}(\bar{x})\|}{\sum_{j=1}^b \|\nabla f_{S_{m_j}}(x_s^k) - \nabla f_{S_{m_j}}(\bar{x})\|}$ for $i = 1, 2, ...b$. Denote $\mathbf{P} = (P_{r_1}, ..., P_{r_b})^\top$
    Resample $\{S_{m_{r_1}}, ..., S_{m_{r_d}}\}$ from S_m^b based on the resampling distribution $\mathbf{P}$
    Compute the grafting gradient as

$$g_{m,b}(x) = \begin{pmatrix} \frac{1}{bP_{S_{m_{r_1}}}} \left(\frac{1}{m} \sum_{i \in S_{m_{r_1}}} \frac{\partial f_i}{\partial x_1}(x) \right) \\ \vdots \\ \frac{1}{bP_{S_{m_{r_d}}}} \left(\frac{1}{m} \sum_{i \in S_{m_{r_d}}} \frac{\partial f_i}{\partial x_d}(x) \right) \end{pmatrix}$$

    Update:

$$x_s^{k+1} = x_s^k - \gamma \left(g_{m,b}(x_s^k) - g_{m,b}(\bar{x}) + \bar{\mu} \right) \triangleq x_s^k - \gamma \tilde{g}_{m,b}^k$$

  end
  Option (a): Set $\bar{x}_s = \sum_{k=0}^{q-1} p_k x_s^k$ and set $x_{s+1}^0 = x_s^q$
  Option (b): Set $\bar{x}_s = x_s^k$ for randomly chosen $k \in \{0, ..., q-1\}$ and set $x_{s+1}^0 = x_s^q$
  Option (c): Set $\bar{x}_s = x_s^q$ and set $x_{s+1}^0 = x_s^q$
end

where γ is the fixed learning rate and μ is the strong convexity constant. Among these three options, we prefer option (c) at the end of the inner loop since it is easier to implement and more intuitive for practitioners applying to real problems. These three options are all indispensable for the convergence analysis of the GGD-WR-SVRG method.

Similar to SVRG, the bi-loop (inner loop and outer loop) structure and the modified grafting gradient $\tilde{g}_{m,b}^k$ are the key ingredients of variance reduction property. Without the outer loop, the noise variance of the modified grafting gradient $\mathbb{E}\|\tilde{g}_{m,b}^k\|^2$ deems to diverge. If $P_{S_{m_i}} = 1/b$ for $S_{m_i} \in S_m^b$, then the noise variance of modified grafting gradient $\mathbb{E}\|\tilde{g}_{m,b}^k\|^2$ equals to the noise variance of mini-batch stochastic variance reduced gradient $\mathbb{E}\|\nabla f_{S_{m_{r_i}}}(x_s^k) - \nabla f_{S_{m_{r_i}}}(\bar{x}) + \bar{\mu}\|^2$ proposed by Johnson and Zhang (2013). This result suggests that since the resampling distribution $\mathbf{P}$ provided in Algorithm 2 is optimal in sense of minimizing $\mathbb{E}\|\tilde{g}_{m,b}^k\|^2$, the noise variance of the modified grafting gradient is further reduced compared with the original mini-batch stochastic variance reduced gradient. It is

also worth noting that using resampling probability given in Algorithm 2 results in a heavier computational burden as we have to calculate *bmd* partial derivatives in one iteration.

As always, the convergence result of GGD-WR-SVRG under strongly-convex assumption is provided at the first place following the analysis provided by Sebbouh et al. (2019).

Theorem 21 *Suppose that the objective function f and the individual loss function f_i are L-smooth and f_i is μ-strongly convex for all $i \in [n]$. When Algorithm 2 is run with option (a), the fixed stepsize $\gamma < 1/16L$ and the update period q, then the iterates generated by the outer loop satisfy*

$$\mathbb{E}\|x_s^q - x^*\|^2 \leq \rho^s(1 + 12L^2\gamma^2 V_q)\mathbb{E}\|x_0^q - x^*\|^2,$$

where

$$\rho = \max\left\{(1-\gamma\mu)^q, \frac{1}{2}\right\}.$$

Theorem 21 indicates that the iterates generated by GGD-WR-SVRG converge to the optimal point x^* at a linear rate since $\rho \leq 1/2$. The bound we obtained here is comparable to those in Johnson and Zhang (2013) and Sebbouh et al. (2019). Denote by $\kappa = L/\mu$ the conditional number of objective function f. We next give the total complexity result of GGD-WR-SVRG method.

Corollary 22 *If we set the stepsize $\gamma = 1/16L$ and the update period $q = n$. Then to achieve ϵ-optimality, the iteration number for the outer loop T should satisfy*

$$T \geq \max\left\{\frac{16\kappa}{n}, 2\right\}\ln\left(\frac{(64 + 3V_q)\mathbb{E}\|x_0^q - x^*\|^2}{64\epsilon}\right)$$

thus the total complexity is

$$2(1+bm)d \cdot \max\{8\kappa, n\}\ln\left(\frac{(64 + 3V_q)\mathbb{E}\|x_0^q - x^*\|^2}{64\epsilon}\right).$$

From Theorem 21 and Corollary 22, we can see that GGD-WR-SVRG method can achieve ϵ-optimality with a pretty large fixed stepsize $1/16L$ compared with the stepsize used in Corollary 5. But this improvement comes with a price which is that we have to evaluate the full gradient $\nabla f(x)$ once at the begining of the outer loop. This evaluation influences the complexity which is now related to the training set size n. In other words, when using a large data set, GGD-WR-SVRG algorithm may take longer time to achieve ϵ-optimality. Supposing that $m \ll q$ and $b \ll q$, then the complexity will be $\mathcal{O}\left((n+\kappa)d\ln(1/\epsilon)\right)$ which is same as the complexity of the original SVRG method. The convergence results can also be obtained in the general convex case.

Theorem 23 *Suppose that the objective function f and the individual loss function f_i are L-smooth. When Algorithm 2 is run with option (b) and the fixed stepsize $\gamma < 1/10L$. Denoting $\hat{x} = \frac{1}{qT}\sum_{s=1}^{T}\sum_{k=0}^{q-1} x_s^k$, then it satisfies*

$$\mathbb{E}\left[f(\hat{x}) - f(x^*)\right] \leq \frac{P^0}{2qT\gamma(1 - 10L\gamma)}$$

where $P^0 = \mathbb{E}\|\bar{x}_0 - x^*\|^2 + 12L\gamma^2 q\mathbb{E}\left[f(\bar{x}_0) - f(x^*)\right]$.

The GGD-WR-SVRG algorithm has a sublinear convergence rate in the general convex case with the fixed stepsize. We can also find that if the outer loop did not exist, then the convex GGD-WR-SVRG would converge at sublinear rate up to some noise level like the convergence results of original GGD methods. This result suggests that the bi-loop structure is responsible for the variance reduction property. From Theorem 23, we can give the total complexity as follows.

Corollary 24 *If we set the stepsize $\gamma = 0.05/L$ and the update period $q = n$, the iteration number of the outer loop T to achieve ϵ-optimality should satisfy*

$$T \geq \max\left\{\frac{40L\mathbb{E}\|\bar{x}_0 - x^*\|^2}{n\epsilon}, \frac{1.2\mathbb{E}\left[f(\bar{x}_0) - f(x^*)\right]}{\epsilon}\right\},$$

then the complexity is

$$nd(1+bm)\cdot\max\left\{\frac{40L\mathbb{E}\|\bar{x}_0 - x^*\|^2}{n\epsilon}, \frac{1.2\mathbb{E}\left[f(\bar{x}_0) - f(x^*)\right]}{\epsilon}\right\}.$$

If the conditional number κ is not ill-conditioned, $b \ll n$ and $m \ll n$, then the complexity result is $\mathcal{O}(nd\ln 1/\epsilon)$ for strongly convex functions and $\mathcal{O}(nd/\epsilon)$ for general convex functions with relatively large stepsize $\gamma = 0.05/L$. It is also worth noting that if we set the fixed stepsize $\gamma = 1/n^{1/2}$ and the update period $q = n$, then from Theorem 23, we can derive an improved bound for the complexity, that is, $\mathcal{O}(n^{1/2}d/\epsilon)$ for $m \ll n$, $b \ll n$. Recall the complexity results for the GGD method with small fixed stepsize which is dependent on ϵ: $\mathcal{O}((d/\epsilon)\ln 1/\epsilon)$ for strongly convex functions and $\mathcal{O}(d/\epsilon^2)$ for general convex functions. We can see that if the training set size n is not extremely large, GGD-WR-SVRG may require less iterations to achieve ϵ-optimality with fixed stepsize. Finally we state the convergence result for non-convex GGD-WR-SVRG.

Theorem 25 *Suppose that the objective function f and the individual loss function f_i are L-smooth, the objective function f is also bounded below by f_{min}. Define a decreasing sequence $\{\eta_k\}_{k=0}^q$,*

$$\eta_k = 3\gamma^2L^3 + \eta_{k+1}(6\gamma^2L^2 + 1 + \tau\gamma), \text{ with } \eta_q = 0, \tag{11}$$

where fixed stepsize γ and constant $\tau > 0$ satisfying

$$\tau > \frac{\eta_0}{1 - 2\gamma L - 4\gamma\eta_0}.$$

When Algorithm 2 is run with option (c) and fixed stepsize, then the iterates generated by Algorithm 2 satisfy

$$\min_{\substack{k=0,1,\ldots,q-1\\ s=1,2,\ldots,T}} \mathbb{E}\|\nabla f(x_s^k)\|^2 \leq \frac{\mathbb{E}\left[f(x_1^0) - f_{min}\right]}{qT\gamma(1 - 2\gamma L - 4\gamma\eta_0 - \eta_0/\tau)}.$$

Theorem 25 indicates that as qT increasing, the minimum expected square L_2-norm of the full gradient can not stay bounded away from zero, which implies that the non-convex GGD-WR-SVRG has a sublinear convergence rate. With proper choice of γ and τ, we can derive the complexity result for non-convex GGD-WR-SVRG.

Corollary 26 *If we set the stepsize* $\gamma = \psi/(Ln^{2/3})$*, constant* $\tau = L/n^{1/3}$ *and update period* $q = \lceil n/7\psi \rceil$*, where* ψ *is a constant satisfying*

$$\psi \leq \min\left\{\frac{1}{12(e-1)}, \frac{1}{4}\left(\frac{2}{n^{2/3}} + \frac{12(e-1)}{n}\right)^{-1}\right\},$$

then to achieve ϵ*-optimality, the iteration number of outer loop* T *should satisfy*

$$T \geq \frac{14L\mathbb{E}\left[f(\bar{x}_0) - f_{min}\right]}{\epsilon n^{1/3}}.$$

Thus the total complexity is

$$d\left(n + \lceil n/7\psi \rceil mb\right) \cdot \frac{14L\mathbb{E}\left[f(\bar{x}_0) - f_{min}\right]}{\epsilon n^{1/3}}.$$

From Corollary 26, we can see that non-convex GGD-WR-SVRG can still achieve ϵ-optimality with a stepsize $\gamma = \psi/(Ln^{2/3})$ which is related to the training set size n. The complexity will be $\mathcal{O}(dn^{2/3}/\epsilon)$ for $m \ll n$, $b \ll n$. Comparing Corollary 26 with Corollary 24, we can find that when convex assumption does not hold, the complexity will degrade from $\mathcal{O}(dn^{1/2}/\epsilon)$ to $\mathcal{O}(dn^{2/3}/\epsilon)$ which is still better than $\mathcal{O}(dn/\epsilon)$, the complexity result for convex GGD-WR-SVRG with fixed stepsize that is independent of n. This result suggests that a stepsize which is dependent on n may be more proper to use in GGD-WR-SVRG algorithm for convex or non-convex cases.

5.2 GGD-WR-Adam Method

Although variance reduced GGD-based method can achieve a linear convergence rate under strongly-convex assumption, it is potentially burdened by the expensive computational cost and the inadequacy that the iterates generated by variance reduction methods are prone to be stuck in the local minima. As mentioned before, adaptive stepsize methods can accelerate the training process and empirically outperform the original SGD method. Luckily, GGD is also compatible with adaptive stepsize methods. In the rest of this section, we propose GGD-WR-Adam which builds upon Adam and replace the stochastic sample gradient with grafting gradient using sampling with replacement. Before diving into the detail of GGD-WR-Adam algorithm, we first give one additional assumption.

Assumption 27 *The* L_∞*-norm of the grafting gradients is uniformly almost sure bounded, i.e., there is a constant* $R \geq \sigma$ *so that*

$$\|g_{m,b}(x)\|_\infty \leq R - \sqrt{\sigma}, \text{ for all } x \in \mathbb{R}^d \text{ a.s..}$$

This assumption is essential to the convergence analysis of GGD-WR-Adam. $\sqrt{\sigma}$ is used to simplify the final bound as remarked in Défossez et al. (2020).

In Algorithm 3, $g_k^2 = g_k \odot g_k$ indicates the element-wise square, and for a sequence of vectors $\{\nu_k\}$, we denote $\nu_{k,(i)}$ the i-th component of k-th vector in this sequence. We also assume that we have an access to the oracle $\text{GGD}_{m,b}$, i.e., the first few steps in GGD-WR algorithm, which can provide i.i.d grafting gradient samples given m, b, and x^k. Good

Algorithm 3: GGD-WR-Adam method

Input: Batch size b, subsampled set size m, learning rate γ, an orcale $\mathrm{GGD}_{m,b}$, exponential decay rate for the first moment estimates β_1, exponential decay rate for the second moment estimates β_2 and σ.

Initialize: $x^0 = 0$, $h_0 = 0$ and $v_0 = 0$.

for $k = 1, 2, ..., T$ **do**

$g_k = \mathrm{GGD}_{m,b}(x^{k-1})$

$h_k = \beta_1 h_{k-1} + g_k$

$v_k = \beta_2 v_{k-1} + g_k^2$

$\gamma_k = \gamma \cdot \frac{1-\beta_1}{(1-\beta_2)^{1/2}} \cdot \frac{(1-\beta_2^k)^{1/2}}{1-\beta_1^k}$

for $i = 1, 2, ..., d$ **do**

Update: $x_{(i)}^k = x_{(i)}^{k-1} - \gamma_k \frac{h_{k,(i)}}{(\sigma + v_{k,(i)})^{1/2}}$

end

end

default settings for the hyperparameters are $b = 2$, $m = 2^k$ for $k \in \mathcal{N}^+$, $\gamma = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\sigma = 10^{-8}$ which is used for numerical stability. Following the analysis provided by (Défossez et al., 2020), we only present the theoretical bound and complexity result for non-convex GGD-WR-Adam method.

Theorem 28 *Suppose that Assumption 27 holds, the objective function f and the individual loss function f_i are L-smooth, the objective function f is bounded below by f_{min}. When Algorithm 3 is run with the fixed stepsize $\gamma > 0$, $0 < \beta_2 < 1$, $0 \leq \beta_1 < \beta_2$, then for any $T \in \mathcal{N}^+$ such that $T > \beta_1/(1-\beta_1)$, the iterates generated by Algorithm 3 satisfy*

$$\mathbb{E}\|\nabla f(x^\omega)\|^2 \leq \frac{2R\mathbb{E}\left(f(x^0) - f_{min}\right)}{\gamma \tilde{T}} + \frac{J}{\tilde{T}}\left(\ln\left(1 + \frac{R^2}{\sigma(1-\beta_2)}\right) - T\ln(\beta_2)\right),$$

where $\tilde{T} = T - \beta_1/(1-\beta_1)$,

$$J = \frac{\gamma d R L}{(1-\beta_1)(1-\beta_2)(1-\beta_1/\beta_2)} + \frac{\gamma^2 d L^2}{2(1-\beta_2)^{3/2}(1-\beta_1)^{5/2}(1-\beta_1/\beta_2)} + \frac{6dR^2}{(1-\beta_1)^{3/2}(1-\beta_2)^{1/2}(1-\beta_1/\beta_2)^3},$$

and ω is a random index taking values from $\{0, 1, ..., T-1\}$ with probability

$$\forall k \in \mathcal{N},\ k < T,\ P(\omega = k) \propto 1 - \beta_1^{T-k}.$$

The bound derived in Theorem 28 is different from the bound given by Défossez et al. (2020) since we do not leave the corrective term for the first moment estimates as the original Adam algorithm. This theoretical bound seems too complicated to acknowledge that GGD-WR-Adam can converge with careful choice of hyperparameters. So we first give the complexity result and then bring out some discussions about these results. If we set

$\gamma = \tilde{\gamma}/\sqrt{T}$, $\beta_2 = 1 - 1/T$, and assuming that $\beta_1/(1-\beta_1) \ll T$ and $\beta_1/\beta_2 \approx \beta_1$ (These two assumptions can easily hold when iteration number T is extremely large), then the theoretical bound given in Theorem 28 can be approximated by

$$\mathbb{E}\left[\left\|\nabla f\left(x^{\omega}\right)\right\|^2\right] \lesssim 2R\frac{\mathbb{E}[f\left(x^0\right) - f_{min}]}{\tilde{\gamma}\sqrt{T}} + \frac{1}{\sqrt{T}}\left(\frac{\tilde{\gamma}dRL}{(1-\beta_1)^2} + \frac{\tilde{\gamma}^2 dL^2}{2(1-\beta_1)^{7/2}} + \frac{6dR^2}{(1-\beta_1)^{9/2}}\right)\left(\ln\left(1 + \frac{TR^2}{\sigma}\right) + 1\right).$$

Denoting

$$K = \left(\frac{\tilde{\gamma}dRL}{(1-\beta_1)^2} + \frac{\tilde{\gamma}^2 dL^2}{2(1-\beta_1)^{7/2}} + \frac{6dR^2}{(1-\beta_1)^{9/2}}\right),$$

then we can obtain

$$\mathbb{E}\left[\left\|\nabla f\left(x^{\omega}\right)\right\|^2\right] \lesssim 2R\frac{\mathbb{E}[f\left(x^0\right) - f_{min}]}{\tilde{\gamma}\sqrt{T}} + \frac{K}{\sqrt{T}}\left(\ln\left(1 + \frac{TR^2}{\sigma}\right) + 1\right). \tag{12}$$

Corollary 29 *To achieve ϵ-optimality of non-convex GGD-WR-Adam method, for some constant $\phi \in (0, 1/2)$, the iteration number T should satisfy*

$$T \geq \max\left\{\frac{36R^2\left[\mathbb{E}f(x^0) - f_{min}\right]^2}{\tilde{\gamma}^2\epsilon^2}, \frac{\sigma}{R^2}\left(\left(\frac{3K}{\phi e\epsilon}\right)^{\frac{2}{1-2\phi}} \cdot \left(1 + \frac{R^2}{\sigma}\right)^{\frac{1}{1-2\phi}} - 1\right)\right\}.$$

Then the total complexity is

$$md \cdot \max\left\{\frac{36R^2\left[\mathbb{E}f(x^0) - f_{min}\right]^2}{\tilde{\gamma}^2\epsilon^2}, \frac{\sigma}{R^2}\left(\left(\frac{3K}{\phi e\epsilon}\right)^{\frac{2}{1-2\phi}} \cdot \left(1 + \frac{R^2}{\sigma}\right)^{\frac{1}{1-2\phi}} - 1\right)\right\}.$$

Now we put some remarks on the theoretical results derived in Theorem 28 and Corollary 29. For $m \ll n$, non-convex GGD-WR-Adam achieves ϵ-optimality albeit with the complexity of $\mathcal{O}(d^{1+2/(1-2\phi)}/\epsilon^{2/(1-2\phi)})$ which is larger than the complexity of non-convex GGD since non-convex GGD-WR-Adam has a slower convergence rate $\mathcal{O}(\ln(T)/\sqrt{T})$. Recalling that we assume that there is a uniform almost sure bound for the L_∞-norm of grafting gradients and apply anisotropic stepsizes to each dimension of model parameters, parameter dimension d is introduced into theoretical bounds and the complexity result is dependent of $d^{1+2/(1-2\phi)}$ as dR^2 is a natural bound for the L_2-norm of grafting gradient. It is also noteworthy that β_1 does not play an important role in the theoretical bounds as it is regarded as a constant that can be absorbed by the increasing iteration number T. However, β_1 serves the purpose of deciding random index ω crucially. For one hand, $\beta_1 > 0$ implies that Algorithm 3 is run with heavy-ball style momentum. The closer β_1 approaches 1, the less momentum h_k changes in one iteration, that is, the last few grafting gradients barely influence the direction of the momentum. So the first few iterations are more likely to be selected since they are cumulated through time and are more important in deciding the direction of momentum for $\beta_1 \to 1$. On the other hand, $\beta_1 = 0$ implies that ω is uniformly picked from $\{0, 1, ..., T-1\}$. This is expected as well since $\beta_1 = 0$ also implies that there is no momentum and every iterate contributes evenly to the direction of update.

Data set	Dim	n_{tr} (train)	Sparsity	n_{te} (test)	$\bar{L}$	$\max L$	κ
covtype	54	290,506	22.22%	290,506	1.2258	1.8921	3.56102×10^5
ijcnn1	22	49,990	59.09%	91,071	0.3763	0.9842	1.88112×10^4
a9a	123	32,561	11.28%	9,865	3.4673	3.5000	1.12898×10^5
rcv1	47,236	20,242	0.1549%	677,399	0.2441	0.2500	4.94107×10^3

Table 1: Summary of data sets

6. Experiment Results

Our empirical results are presented in this section. We evaluate the performance of grafting gradient based algorithms on solving strongly-convex and non-convex problems, and compare their performance with vanilla SGD, SGD with importance sampling, mini-batch SGD, variance reduction method SVRG and adaptive stepsize method Adam.

6.1 Binary Classification Problems

We first run experiments on the L_2-regularized logistic regression problem given by

$$f_i(x) = -\left(b_i \ln\left(\frac{1}{1+\exp^{-a_i^\top x}}\right) + (1-b_i)\ln\left(\frac{\exp^{-a_i^\top x}}{1+\exp^{-a_i^\top x}}\right)\right) + \frac{\lambda}{2}\|x\|^2,$$

where $(a_i, b_i) \in \mathbb{R}^d \times \{0,1\}$, $i = 1, ..., n_{\text{tr}}$ are the data samples from *covtype*, *ijcnn1*, *a9a* and *rcv1*. All data sets are available on `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, and are widely used in other literatures (Nguyen et al., 2017; Qian et al., 2019; Sebbouh et al., 2019; Mishchenko et al., 2020; Huang et al., 2021; Malinovsky et al., 2021). Relevant statistics of data and loss function are summarized in Table 1.

In Table 1, Dim denotes the features number of the training data, n_{tr}, n_{te} are the number of data used for training and testing respectively, sparsity is the proportion of non-zero values in training data features. $\bar{L}$ is the average of smoothness constants, $\max L$ is the maximum of smoothness constants and κ is the conditional number of objective function f. They can be calculated explicitly since the loss function of L_2-regularized logistic regression problem is μ-strongly convex with $\mu = \lambda$ and L_i-smooth which admits a closed form expression $L_i = \|a_i\|^2/4 + \lambda$.

For *ijcnn1*, *a9a* and *rcv1*, we use the predefined training set and testing set. *covtype* does not have a testing set. In that case, we randomly split the data set into the training set and the testing set with 50% for training and 50% for testing.

The penalty parameter λ is set to be $1/n_{\text{tr}}$ for all the experiments on different data sets. Since the stepsizes for stochastic optimization method are critical, we adopt the popular t-inverse learning schedule $\gamma_k = \gamma_0(1+\gamma_d\lfloor k/n_{\text{tr}}\rfloor)^{-1}$ (Johnson and Zhang, 2013; Reddi et al., 2016), where k is the iteration number and γ_0, γ_d are chosen so that the corresponding algorithms give the best performance. When a fixed stepsize is used, we set $\gamma_d = 0$. For the methods which use grafting gradients to update the parameters in one iteration, unless specifying, the size of subsampled set m is set to be 16 for *ijcnn1*, *a9a* and *rcv1* and 256 for *covtype* since its size is way bigger than other training sets. The batch size b used in the

grafting gradient based methods is set to be 2 since $b = 2$ gives the best performance and saves the most computational cost.

In the practical implementation, since most machine learning libraries calculate partial derivatives through backpropagation and the chain rule, to construct the grafting gradient, we actually obtain the entire $b \times m \times d$ partial derivatives through backpropagation and discard the unused partial derivatives. This implementation implies that in one iteration, the actual computational complexity of GGD is approximately b times larger than that of mini-batch SGD with the same mini-batch size m, and $b \times m$ times larger than that of vanilla SGD. So for fair comparison, the total iteration number of vanilla SGD and SGD with importance sampling is set to be $b \times m$ larger than that of GGD and the mini-batch size of mini-batch SGD, SVRG and Adam is set to be $b \times m$ in all these experiments to maintain the total computational complexity at the same level.

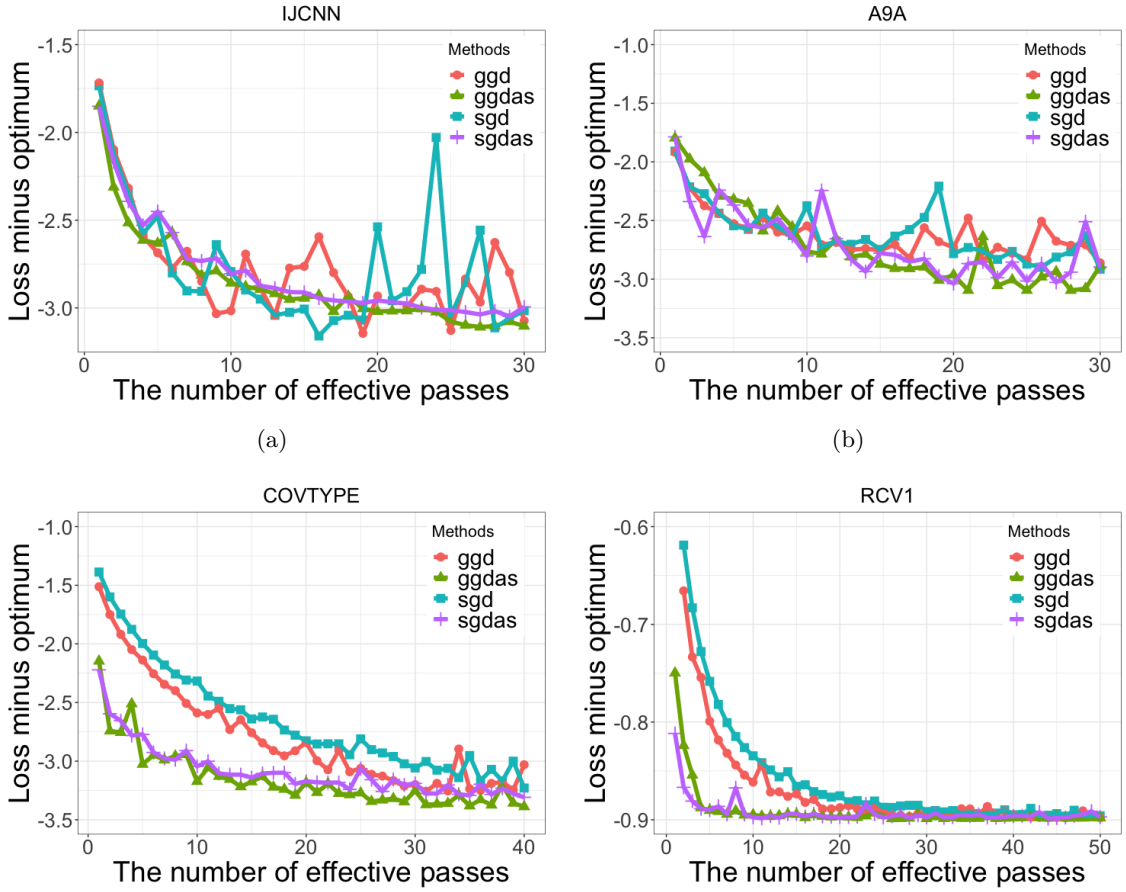


Figure 2: Comparisons of the train loss between vanilla SGD and GGD methods on ijcnn1, a9a, covtype and rcv1.

Comparison between vanilla SGD and GGD. In all these experiments, we compare the performances of different methods in terms of train loss, the square L_2-norm of the full gradient and the test loss (results are listed in Appendix). The horizontal axis of these figures denotes the number of effective passes. Usually, one effective pass over data is considered as computing one full gradient or evaluating n_{tr} times gradients (total $n_{\text{tr}\times d}$ partial derivatives). Suffix *-as* means that the diminishing stepsize sequences are used in the algorithm. Take a deep look at Figure 2. Compared with the stochastic sampled gradient, the improvements brought by the grafting gradient mainly lie in two aspects: One is that the iterates generated by GGD can sometimes decrease faster than vanilla SGD and the other is that the iterates can fluctuate in a small neighborhood of the optimum point. For the former one, since GGD which uses a batch of subsampled sets to update the parameters, compared with the best-tuned SGD, the best-tuned GGD can fit a larger fixed stepsize which possibly results in a faster convergence as shown in Figures 2(a), 2(c) and 2(d). For the latter

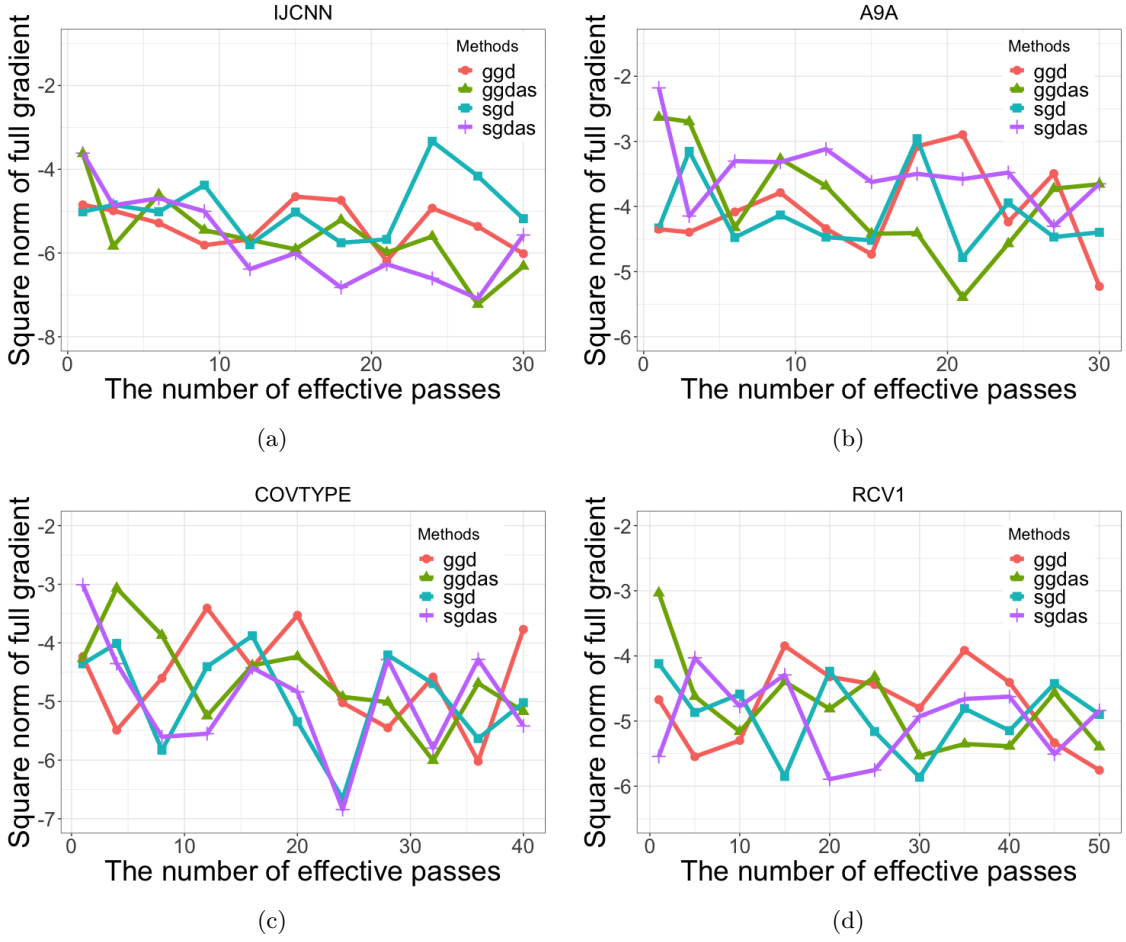


Figure 3: Comparisons of the square L_2-norm of full gradient $\|\nabla f(x^k)\|^2$ between vanilla SGD and GGD methods on ijcnn1, a9a, covtype and rcv1.

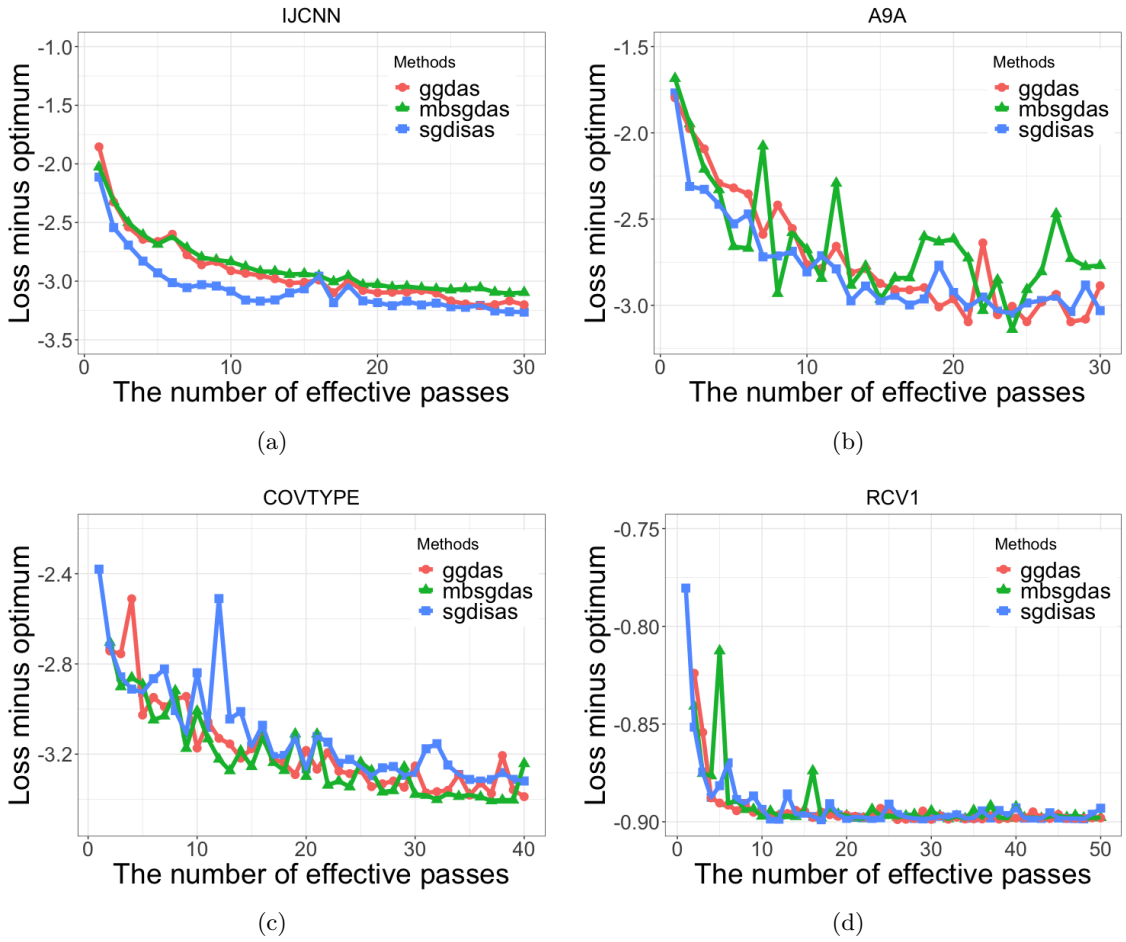


Figure 4: Comparisons of the train loss between GGD-as, mini-batch SGD-as and SGD-as with importance sampling methods on ijcnn1, a9a, covtype and rcv1.

one, it coincides with the theoretical result derived in Theorem 4 that the noise variance of grafting gradient is reduced by a constant factor. Figure 3 presents the comparison in terms of $\|\nabla f(x^k)\|^2$. We can observe that although the comparison is less clear, GGD with diminishing stepsize sequence still outperforms its competitors in Figures 3(a) and 3(b) as it achieves a lower value of $\|\nabla f(x^k)\|^2$, which implies that the iterates generated by GGD with diminishing stepsize sequence are more closer to the optimum point due to the strong convexity of the objective function. From Figures 3(c) and 3(d), we can see after about 10 epoches of training, the iterates generated by GGD with diminishing stepsize sequence fluctuate more smoothly compared with its competitors, which demonstrate the variance reduction property possessed by GGD method. In a short word, these results suggest that using GGD can be more beneficial than using vanilla SGD for solving L_2-regularized logistic regression problems.

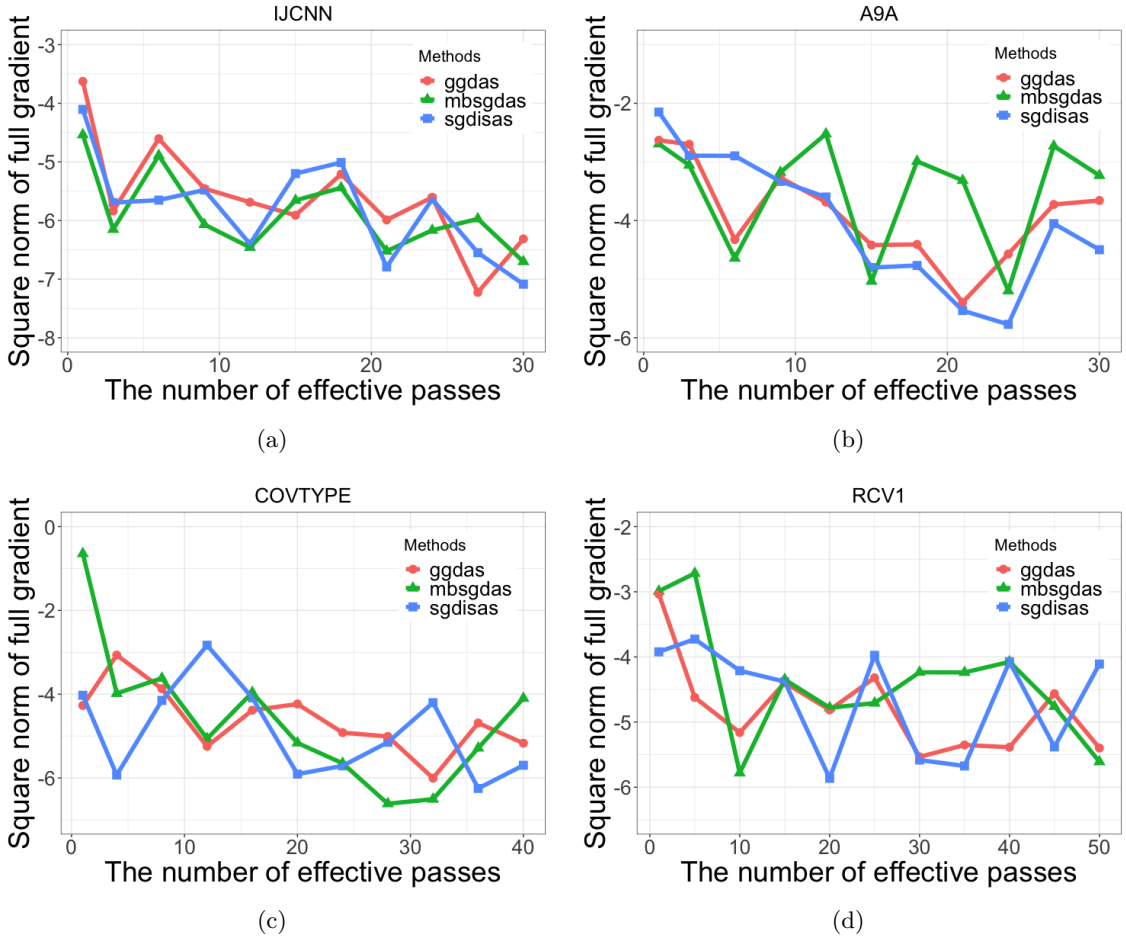


Figure 5: Comparisons of the square L_2-norm of full gradient $\|\nabla f(x^k)\|^2$ between GGD-as, mini-batch SGD-as and SGD-as with importance sampling methods on ijcnn1, a9a, covtype and rcv1.

Comparison among GGD, MBSGD and SGD-IS. Figures 4 and 5 report the performance of GGD, mini-batch SGD (mbsgd) and SGD with importance sampling (sgdis). SGD-IS is run without minibatching, that is, using one gradient sampled from population with probability $P_{r_i} \propto L_{r_i}$. For the performances of mini-batch SGD with importance sampling, they are presented in Appendix C. In view of the algorithms with fixed stepsize being inferior to the algorithms with diminishing stepsize sequence, we do not report the performance of GGD, MBSGD and SGD-IS with fixed stepsize.

It is clear that although SGD-IS may outperform MBSGD as shown in Figures 4(a), 4(b) and 5(b), and vice versa such like Figure 4(c), the performances of GGD algorithm with diminishing stepsize sequence are pretty robust and close to the better one out of MBSGD and SGD-IS. For example in Figure 4(a), GGD, MBSGD and SGD-IS show a same decreasing rate for the first few epoches, but they begin to differ after three epoches.

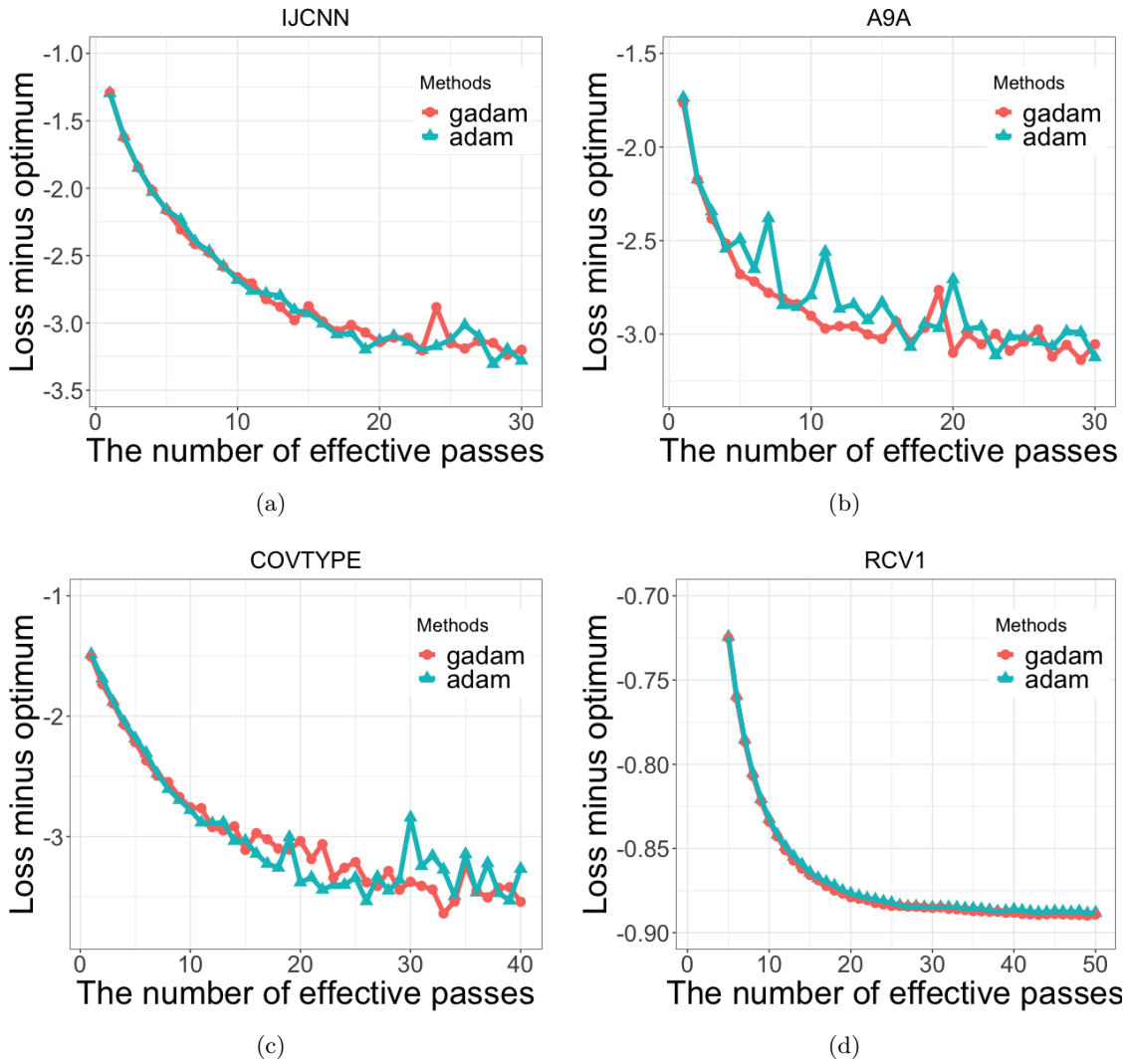


Figure 6: Comparisons of the train loss between GGD-WR-Adam and Adam methods on ijcnn1, a9a, covtype and rcv1.

SGD-IS decreases faster than MBSGD and GGD, and eventually achieves a lower train loss. Although GGD performs quite similarily to MBSGD in the first few epoches, it gradually outperforms MBSGD and catches up with SGD-IS. Robustness can also be confirmed by Figure 4(c) where MBSGD and GGD decrease in a much stable manner compared with SGD-IS. These results confirm the doubly robust property which we have proven in Section 4 and empirically show that GGD can obtain robust and comparable results when training the logistic regression models with L_2-regularization.

Comparison between Adam and GAdam. To compare the performance of Adam and GGD-WR-Adam (gadam) methods, we use minibatching technique to update the parameters both for Adam and GAdam, set fixed learning rates and the hyperparameters β_1,

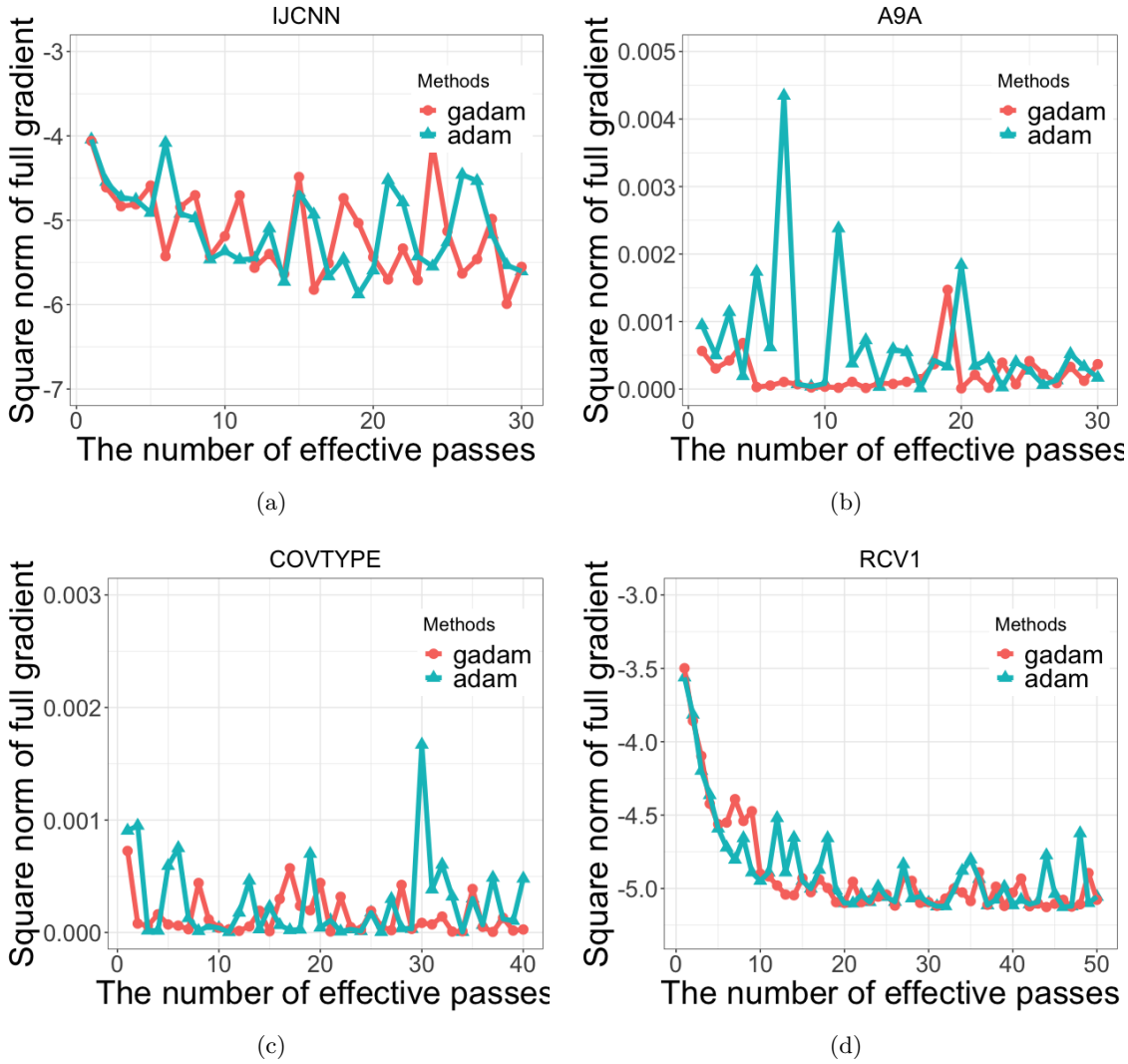


Figure 7: Comparisons of the square L_2-norm of full gradient $\|\nabla f(x^k)\|^2$ between GGD-WR-Adam and Adam methods on ijcnn1, a9a, covtype and rcv1.

β_2 and σ are set by default for these two methods. From the results in Figure 6, we can see that in most cases, the difference between the performances of Adam and GAdam is quite nuanced especially for *rcv1* data set where descent curves overlap each other. GAdam and Adam both show a similar decreasing rate and achieve a low level of train loss except for *a9a* data set. In Figure 6(b), the iterates of GAdam algorithm decrease in a much stable manner compared with Adam method. Recalling that SGD-IS empirically outperforms MBSGD on *a9a* data set, we can infer that since using importance sampling technique may be more beneficial than using mini-batching technique for training L_2-regularized logistic regression model on *a9a* data set.

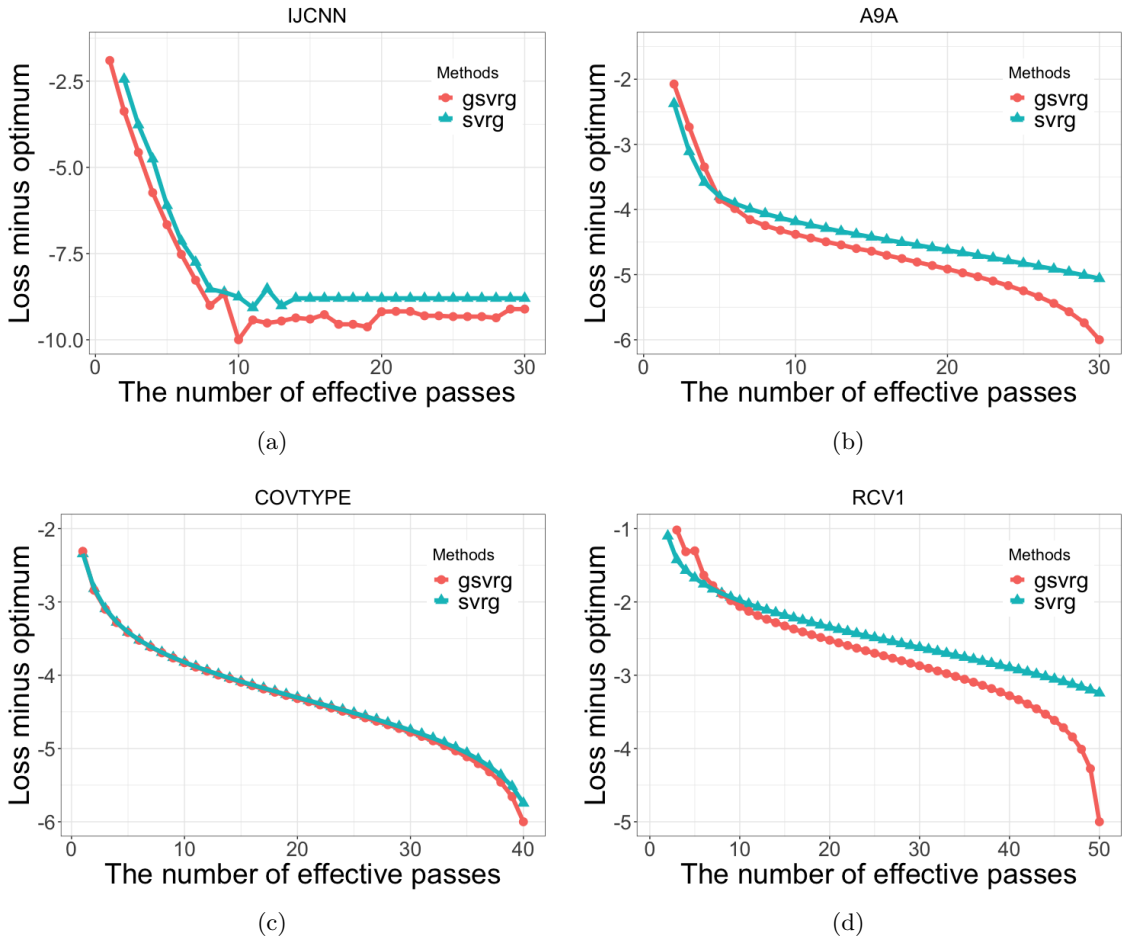


Figure 8: Comparisons of the train loss between GGD-WR-SVRG and SVRG methods on ijcnn1, a9a, covtype and rcv1.

From Figures 7(b), 7(c) and 7(d), we can see that in terms of $\|\nabla f(x^k)\|^2$, the iterates generated by Adam fluctuate more greatly than the iterates generated by GAdam. These results indicate that empirically the noise variance of GAdam may be smaller than the noise variance of Adam.

Comparison between SVRG and GSVRG. In these experiments, GGD-WR-SVRG (gsvrg) is competing against the mini-batch SVRG. The update period q is set to be $\lfloor 0.26n \rfloor$ for *ijcnn1*, $\lfloor 1.95n \rfloor$ for *a9a*, $\lfloor 0.62n \rfloor$ for *covtype* and $\lfloor 0.14n \rfloor$ for *rcv1* as suggested in Sebbouh et al. (2019). We can see that GSVRG outshines the original mini-batch SVRG method except for *covtype* data set where GSVRG only holds a slender lead and the improvement is pretty much negligible. Results in Figure 8 show that GSVRG can achieve a lower train loss compared with SVRG method for *ijcnn* data set. For *a9a* and *rcv1* data sets, although the performances of GSVRG are a liitle worse than the performances of SVRG at the beginning,

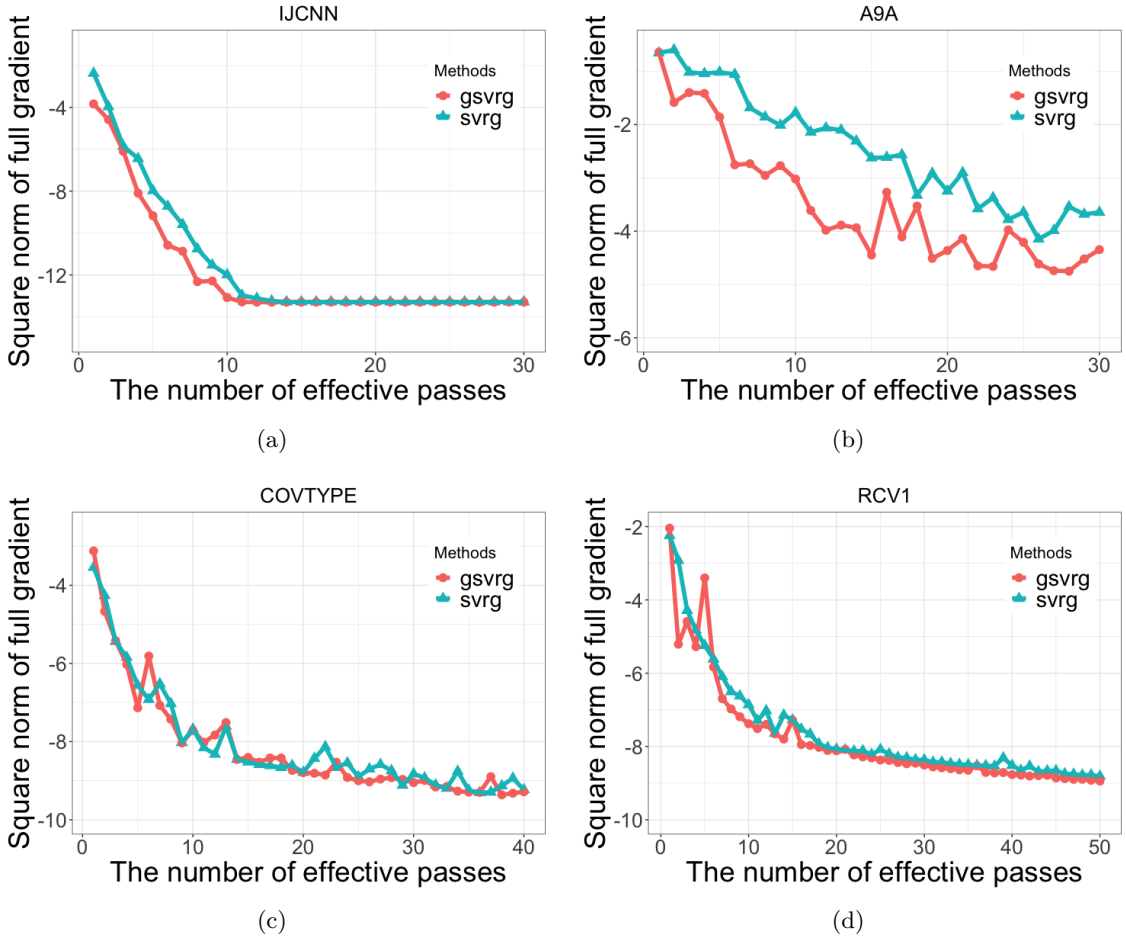


Figure 9: Comparisons of the square L_2-norm of full gradient $\|\nabla f(x^k)\|^2$ between GGD-WR-SVRG and SVRG methods on ijcnn1, a9a, covtype and rcv1.

the decreasing rate of GSVRG gradually catches up with SVRG and becomes a bit faster than SVRG in the last few epoches.

In terms of $\|\nabla f(x^k)\|^2$, there is no significant difference between the performances of GSVRG and SVRG except that the comparison in *a9a* data set is quite discernible. Results in Figures 4(b), 5(b), 7(b) and 9(b) suggest that methods, which assign non-uniform sampling probability to data sample, may outperform these with uniform sampling probability for training L_2-regularized logistic regression model on *a9a* data set.

6.2 Multiclass Classification Problems

We then run experiments to solve the multiclass classification problems via training convolution neural networks which is one representative non-convex problem encountered in machine learning. We use two common data sets, *MNIST* (LeCun et al., 2010) and *CIFAR-10* (Krizhevsky et al., 2009) to train two convolution neural networks with different structures.

For the former one, we train the classic LeNet-5 (LeCun et al., 1998) with minimal modification, which consists of two convolution layers with batch normalization and relu activation function, two fully-connected layer with relu activation function and one fully-connected layer with softmax activation function to output the predicted values. For the latter one, we use the cifar10-nv architecture proposed by Gitman and Ginsburg (2017) which achieves close to the state-of-the-art performance in less training time. The complete network architectures are presented in Appendix C. L_2-regularization are used for preventing overfitting in these experiments and the penalty parameter λ is set to be 10^{-4}. Features in data sets are normalized to the interval $[0, 1]$ for all the experiments and the images from *MNIST* are resized into 32×32 to fit the LeNet-5 architecture. Since mini-batch SGD is much more efficient than vanilla SGD in neural network training, we do not present the result of vanilla SGD methods. For the convolution neural network training, the L-smoothness constant of individual loss function is not available and hence the result of SGD with importance sampling is not provided either. In previous experiments, we empirically show that the methods with diminishing stepsize sequence converge much better than those methods with fixed stepsize. Hence only the algorithms with diminishing stepsize sequence are compared in solving multiclass classification problems.

We train all the networks with the subsampled set size $m = 128$ for GGD based methods. For fair comparison, the mini-batch size is set to be 256 for mini-batch SGD, SVRG and Adam as $b = 2$. We adopt the linear decay learning rate which is defined as

$$\gamma_k = \gamma_0 + \frac{\gamma_{T-1} - \gamma_0}{T - 1} \cdot k,$$

where γ_{T-1} is the final learning rate, γ_0 is the initial learning rate, $T-1$ is the total number of effective passes and k is the current number of effective passes. The final learning rate is 10^{-5} and the initial learning rate is 10^{-2} for MBSGD and GGD. The final learning rate is 10^{-6} and the initial learning rate is 10^{-4} for Adam and GAdam. As suggested in Section 5, the final learning rate is $1/n_{\text{tr}}^{2/3}$ and the initial learning rate is $100/n_{\text{tr}}^{2/3}$ for SVRG and GSVRG methods. The batch size b for grafting gradient based methods is set to be 2 and the update period for variance reduction methods q are set to be $3n_{\text{tr}}/m$. Since Katharopoulos and Fleuret (2018); Johnson and Guestrin (2018); Müller et al. (2019) all show that putting non-uniform sampling probability on data samples can benefit the neural network training, we hope that grafting gradient based methods can outperform stochastic gradient based methods for training CNNs. Experiment results are presented in Figures 10 and 11.

From Figure 10, we can see that due to the importance resampling technique, train loss obtained by GGD decreases in a less fluctuating way especially at the beginning of the training process when the relatively large stepsizes are used. As the number of effective passes grows, GGD eventually outperforms and hold a considerable lead over MBSGD. The comparison between GAdam and Adam is more obvious as Adam does not procede GAdam during the entire training process. As for the comparison between GSVRG and SVRG, although result in Figure 10(e) indicates that the performances of GSVRG and SVRG are quite indistinguishable before 80 epoches, GSVRG gradually emerges as a more competitive stochastic optimization method for training LeNet-5 on *MNIST* data set near the end of training process. In conclusion, GGD, GAdam and GSVRG all achieve a lower train loss compared with MBSGD, Adam and SVRG respectively.

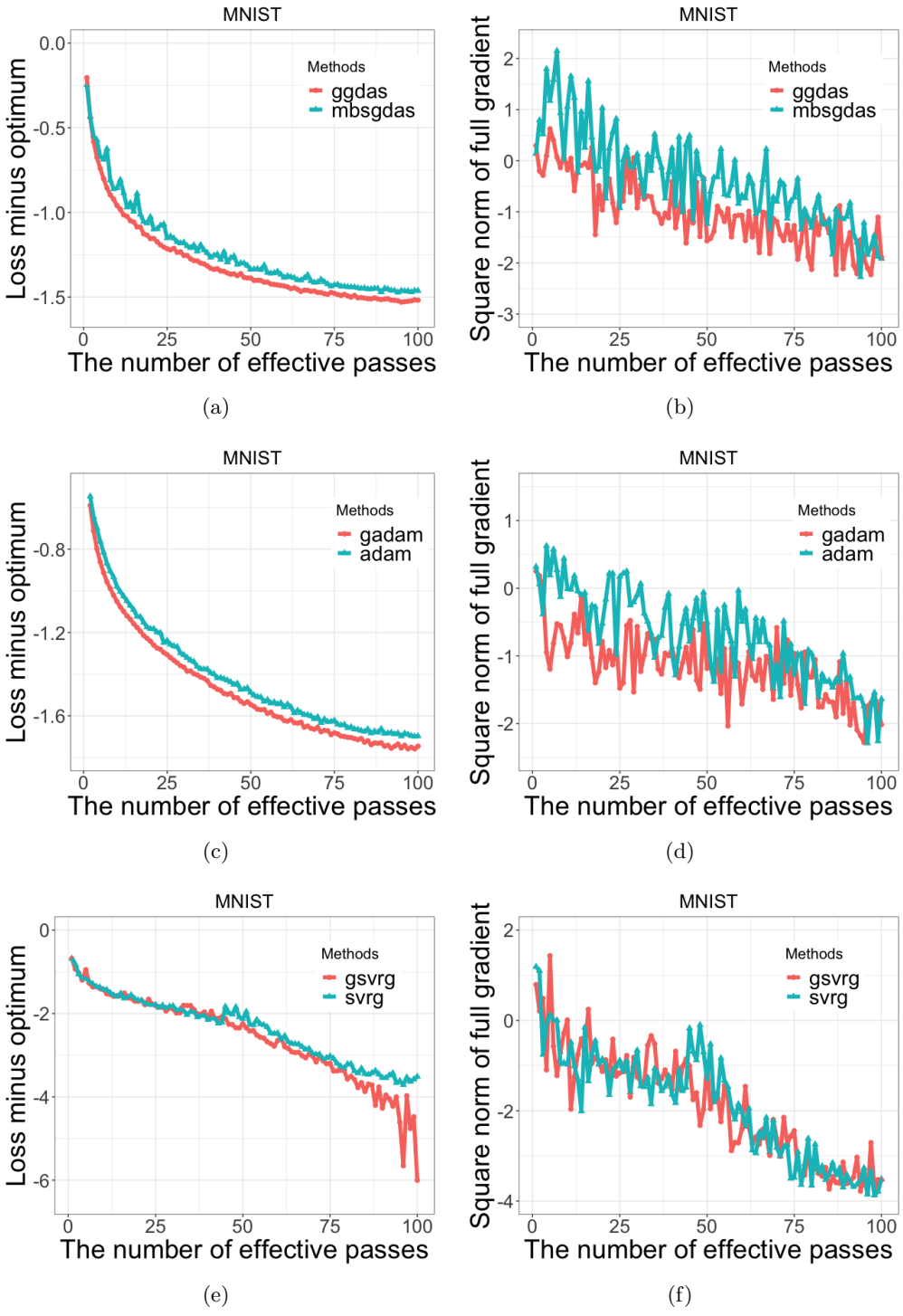


Figure 10: Comparisons of the train loss (left) and the square L_2-norm of full gradient $\|\nabla f(x^k)\|^2$ (right) between MBSGD and GGD, Adam and GAdam, SVRG and GSVRG methods on MNIST.

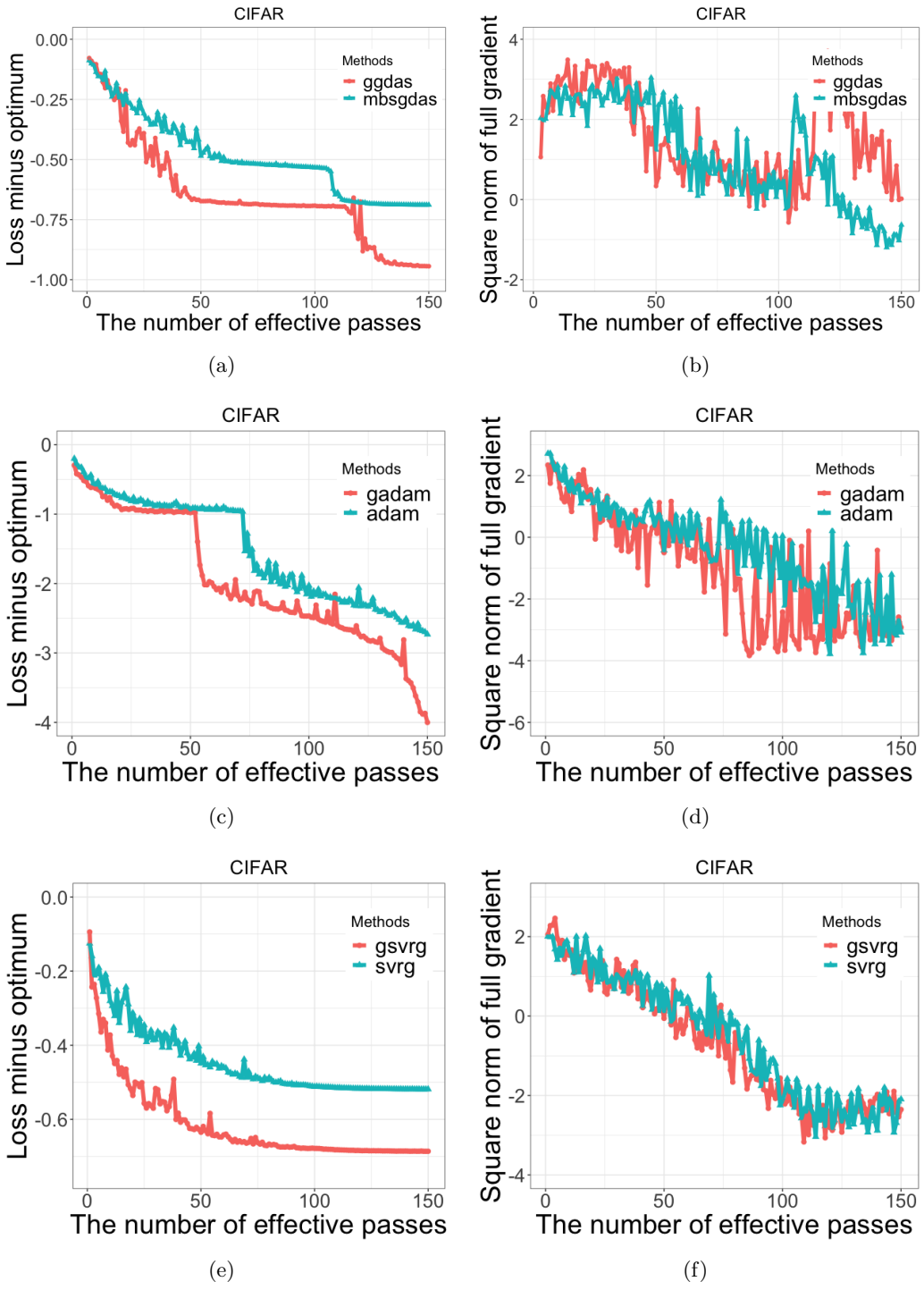


Figure 11: Comparisons of the train loss (left) and the square L_2-norm of full gradient $\|\nabla f(x^k)\|^2$ (right) between MBSGD and GGD, Adam and GAdam, SVRG and GSVRG methods on CIFAR-10.

Methods	Adam	GAdam	MBSGD	GGD	SVRG	GSVRG	scale
Values	5.1588	5.2779	5.3185	5.8986	0.13407	0.15203	$\times 10^{-3}$

Table 2: Minimum square L_2-norm of full gradient on MNIST

Methods	Adam	GAdam	MBSGD	GGD	SVRG	GSVRG	scale
Values	1.6565	1.4659	634.79	2681.0	2.0849	3.7647	$\times 10^{-4}$

Table 3: Minimum square L_2-norm of full gradient on CIFAR-10

For *CIFAR-10* data set, it is intriguing if we take a deep look at Figures 11(a) and 11(c), we will find that the descent curves of objective function values have a same pattern, that is, they all move down slowly, become flattened and slump at some epoches. Figures 11(b) and 11(d) also reflect this descent pattern to a certain extent as $\|\nabla f(x^k)\|^2$ becomes more fluctuated before and after the epoches where the slump happens. In our opinion, the cause behind this descent pattern may be the complexity of network architecture. As for the comparison between different methods, Figure 11(a) shows that after the first few epoches, GGD with linear decay learning rate outperforms MBSGD completely. Even for some train losses of the GGD method before the slump, they are slightly smaller than the minimum train loss found by MBSGD. The comparisons between Adam and GAdam, SVRG and GSVRG are more clear as the descent curves of Adam and SVRG lie above the descent curves of GAdam and GSVRG almost entirely during the training process. Another interesting fact is that although GSVRG and SVRG has a much lower $\|\nabla f(x^k)\|^2$ as expected, the lowest train loss is obtained by the adpative stepsize method, GAdam. This result suggests that one critical problem with the variance reduction methods in minimizing non-convex objective functions is that due to the lack of randomness, the iterates generated by variance reduction methods are likely to be trapped in local minimum. It may also explain why there is no slump in Figure 11(e).

It is worth noting that the comparisons in terms of $\|\nabla f(x^k)\|^2$ are both ambiguous on these two data sets except for Figures 10(b) and 10(d) where grafting gradient based methods obtain a lower $\|\nabla f(x^k)\|^2$ for the most time. Recalling that we derive the theoretical bounds for $\min \mathbb{E}\|\nabla f(x^k)\|^2$ under non-convex assumption, these minimum values are also reported in Tables 2 and 3 for comparison. From these results, we can see that although the minimum values obtained by grafting gradient based methods are larger than that of stochastic sampled gradient based methods (MBSGD, Adam, SVRG), the train losses obtained by grafting gradient based methods are lower. In other words, stochastic sampled gradient based methods may converge much closer to some stationary points and grafting gradient based methods may converge less closer to some stationary points but with a lower objective function value. Combining all these empirical results, we can conclude that using grafting gradient to update the parameters is more robust and promising for empirical risk minimization. Moreover, for training CNNs on *MNIST* and *CIFAR-10* data sets, introducing importance resampling technique can further improve the performance of original stochastic optimization methods.

7. Conclusions and Discussions

We propose a novel stochastic optimization method which employs importance resampling and constructs grafting gradient to update the model parameters. Based on the different sampling techniques, GGD-WR and GGD-WoR are proposed. For the former one, we prove that the grafting gradient using sampling with replacement possesses a doubly robust property which ensures that the performance of GGD-WR will fall between the performances of mini-batch SGD and SGD with importance sampling in sense of expectation. For the latter one, we show that GGD-WoR can be regarded as a more generalized stochastic optimization framework since it includes vanilla SGD, mini-batch SGD and SGD with importance sampling as special cases. Under different assumptions, we provide the convergence analysis for GGD-WR and GGD-WoR methods. Compared with the vanilla SGD method, GGD reduces the noise variance by a constant factor and has a better performance both theoretically and empirically. Compared with mini-batch SGD and SGD with importance sampling, results in Sections 4 and 6 show the unique robustness property possessed by GGD methods. Based on the grafting gradient, we further combine it with high-level variance reduction technique and adaptive stepsize method to improve upon the original GGD methods. The theoretical results of GGD-WR-SVRG and GGD-WR-Adam are presented and the empirical results of GGD-WR-SVRG and GGD-WR-Adam show that they are doing great jobs for solving strongly-convex or non-convex problems. It is worth noting that the performances of coordinate descent and its variants are not compared in Section 6 as they are less relevant to GGD in two counts. One is that grafting gradient updates the parameters in a like manner as SGD and its variants because they update all the parameters in one iteration instead of updating one parameter while keeping all other fixed. On the other hand, a CD is not guaranteed to converge when applied to minimize any given continuously differentiable function. Powell (1973) gave an example of a non-convex continuously differentiable function of three variables where a cyclic CD can not converge to a solution. On the contrary, gradient based methods, including GGD, SGD even gradient descent are guaranteed convergence to a stationary point when objective function is non-convex. Hence we only compare GGD with the most relevant methods, that is, vanilla SGD, mini-batch SGD and SGD with importance sampling in Section 6.

The grafting gradient based methods can be improved in several directions. First, for GGD-WR-SVRG, the optimal resampling probability given in Algorithm 2 is more computational expensive than the resampling probability given in Algorithm 1. It will be more satisfactory if an approximate resampling probability which is defined in terms of objective function values instead of gradient function is adopted in Algorithm 2 with provable theoretical guarantees. Second, although GGD-WR-SVRG empirically outperforms the original SVRG, the theoretical bound derived in Theorem 21 is worse than that of mini-batch SVRG for any values of update period and stepsize. This contradiction indicates that the theoretical bound of GGD-WR-SVRG may be further improved by some technical tricks. Third, GGD-WR-SVRG and SVRG both show that they are less likely to escape from the stationary point. It will be more favorable if some modifications can be made to the procedure of GGD-WR-SVRG so that the additional randomness injected by importance resampling can help the iterates escape from the stationary point.

An promising extension of GGD may be the implementation of GGD in federated learning. In the centralized federated learning (Blanchard et al., 2017; Yin et al., 2018), a trustworthy parameter central server is used to orchestrate all the participating clients and update the parameters with gradients received from clients. Due to the limits in network bandwidth and computing power, sending a complete gradient may be time-consuming. Hence intuitively using grafting gradient update in central parameter server may be more friendly to those clients with low-quality device. The convergence results of "federated" GGD require further discussions as one fundamental challenge in federated optimization method is the presence of non-IID data.

Acknowledgments

The authors thank the action editor Prof. Moritz Hardt and the referee for their valuable comments that greatly improved the presentation of the paper. This work was partial supported by the National Natural Science Foundation of China (12131001), the Fundamental Research Funds for the Central Universities, LPMC, and KLMDASR.

Appendix A. Proofs of Theorems and Corollaries in Section 4

In this appendix we prove the following theorems and corollaries in Section 4. We first prove one useful lemma which are important to the convergence analysis.

Lemma 30 *(Nesterov, 2003) If the function f is L-smooth, then*

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2}\|x - y\|^2, \tag{13}$$

and

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2, \tag{14}$$

where $\langle x, y \rangle = x^\top y$ denotes the inner product of two vectors x and y.

A direct result of (13) is that if we let $x = y - \|\nabla f(y)\|/L$ and assume that f is bounded below by f_{min}, then

$$f_{min} \leq f(y - \|\nabla f(y)\|/L) \leq f(y) - \frac{\|\nabla f(y)\|^2}{L} + \frac{L}{2}\frac{\|\nabla f(y)\|^2}{L^2}.$$

Rearranging the terms, we have

$$\|\nabla f(y)\|^2 \leq 2L\left(f(y) - f_{min}\right), \tag{15}$$

which is quite useful in our convergence analysis. The following Lemma 31 states the relationship between smoothness constant and strong convexity constant, which is useful in the proof of GGD-WR-SVRG.

Lemma 31 *If the function f is both L-smooth and μ-strongly convex, then we have $L \geq \mu$.*

Proof From Definition 1, if x^* is the global minimizer of f, we have

$$f(x) \geq f(x^*) + \frac{\mu}{2}\|x - x^*\|^2.$$

Due to the L-smoothness of f, from Theorem 2.1.5 in Nesterov (2003), we know that

$$f(x) - f(x^*) \leq \frac{L}{2}\|x - x^*\|^2.$$

Combining these two results, we have

$$\frac{\mu}{2}\|x - x^*\|^2 \leq f(x) - f(x^*) \leq \frac{L}{2}\|x - x^*\|^2.$$

We prove Lemma 31 as $\|x - x^*\|^2$ is non-negative. ■

A.1 Proof of Theorem 4

Proof Unless specifying, we write $\mathbb{E}[\cdot \mid x^k]$ as $\mathbb{E}[\cdot]$ for convenience throughout the rest of this paper. We first show that the grafting gradient $g_{m,b}(x^k)$ is an unbiased estimator of $\nabla f(x^k)$ conditional on x^k. Without loss of generality, we take a look at the first dimension of the grafting gradient $g_{m,b}(x^k)$,

$$\begin{aligned}
\mathbb{E}\frac{1}{bP_{S_{m_{r_1}}}}\frac{\partial f_{S_{m_{r_1}}}}{\partial x_1}\left(x^k\right) &= \mathbb{E}\left[\mathbb{E}\left[\frac{1}{bP_{S_{m_{r_1}}}}\frac{\partial f_{S_{m_{r_1}}}}{\partial x_1}\left(x^k\right) \mid S_m^b\right]\right] \\
&= \mathbb{E}\left[\frac{1}{b}\sum_{i=1}^{b}\frac{1}{P_{S_{m_i}}}\frac{\partial f_{S_{m_i}}}{\partial x_1}\left(x^k\right)\cdot P_{S_{m_i}}\right] = \mathbb{E}\left[\frac{1}{b}\sum_{i=1}^{b}\frac{\partial f_{S_{m_i}}}{\partial x_1}\left(x^k\right)\right] \\
&= \mathbb{E}\left[\frac{\partial f_{S_{m_1}}}{\partial x_1}\left(x^k\right)\right] = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial f_i}{\partial x_1}\left(x^k\right),
\end{aligned}$$

which proves that the grafting gradient is an unbiased estimator of $\nabla f(x^k)$ since $\mathbb{E}g_{m,b}(x^k) = \nabla f(x^k)$. The fourth equality holds because $\{S_{m_1}, ..., S_{m_b}\}$ are sampled independently from D_m. For the most gradient-based stochastic optimization algorithms, it makes sense to study the following recursion,

$$\begin{aligned}
\mathbb{E}\|x^{k+1} - x^*\|^2 &= \mathbb{E}\|x^k - x^*\|^2 - 2\gamma\langle x^k - x^*, \nabla f(x^k)\rangle + \gamma^2\mathbb{E}\|g_{m,b}(x^k)\|^2 \\
&\leq (1-\mu\gamma)\|x^k - x^*\|^2 - 2\gamma\left(f(x^k) - f(x^*)\right) + \gamma^2\mathbb{E}\|g_{m,b}(x^k)\|^2. \qquad (16)
\end{aligned}$$

The first inequality relies on the strong convexity of objective function. The last term in (16) can be rewritten as

$$\begin{aligned}
\mathbb{E}\|g_{m,b}(x^k)\|^2 &= \mathbb{E}\left[\frac{1}{b^2}\sum_{i=1}^{b}\frac{1}{P_{S_{m_i}}}\|\nabla f_{S_{m_i}}(x^k)\|^2\right] \\
&= \mathbb{E}\left[\frac{1}{b^2}\left(\sum_{i=1}^{b}\frac{\|\nabla f_{S_{m_i}}(x^k)\|^2}{f_{S_{m_i}}(x^k) - f_{S_{m_i},min}}\right)\left(\sum_{i=1}^{b} f_{S_{m_i}}(x^k) - f_{S_{m_i},min}\right)\right] \\
&= \mathbb{E}\left[\frac{1}{b^2}\left(\sum_{i=1}^{b}\|\nabla f_{S_{m_i}}(x^k)\|^2\right)\right] \\
&+ \mathbb{E}\left[\frac{1}{b^2}\left(\sum_{p\neq q}^{b}\frac{\|\nabla f_{S_{m_q}}(x^k)\|^2}{f_{S_{m_q}}(x^k) - f_{S_{m_q},min}}\left(f_{S_{m_p}}(x^k) - f_{S_{m_p},min}\right)\right)\right] \\
&= \frac{1}{b}\mathbb{E}\|\nabla f_{S_{m_1}}(x^k)\|^2 + \frac{b-1}{b}\mathbb{E}\left[\frac{\|\nabla f_{S_{m_q}}(x^k)\|^2}{f_{S_{m_q}}(x^k) - f_{S_{m_q},min}}\left(f_{S_{m_p}}(x^k) - f_{S_{m_p},min}\right)\right] \\
&\leq \frac{1}{b}\mathbb{E}\|\nabla f_{S_{m_1}}(x^k)\|^2 + \frac{b-1}{b}\mathbb{E}\left[2L_{S_{m_1}}\right]\mathbb{E}\left[f_{S_{m_1}}(x^k) - f_{S_{m_1},min}\right]. \qquad (17)
\end{aligned}$$

The first inequality holds since (15) and subsets are sampled independently. Now we turn to deal the last term in (17) respectively. Noting that $\mathbb{E}\|\nabla f_{S_{m_1}}(x^k)\|^2$ is the noise variance of mini-batch SGD, we first derive an upper bound for this noise variance.

Lemma 32 *For a subset $S_{m_1} \in D_m$. Given x^k, we have*

$$\mathbb{E}\left[\|\frac{1}{m}\sum_{j\in S_{m_1}}^{m}\nabla f_j(x^k)\|^2\right] = \frac{n-m}{m(n-1)}\cdot\frac{1}{n}\sum_{j=1}^{n}\|\nabla f_j(x^k)\|^2 + \frac{n(m-1)}{m(n-1)}\|\nabla f(x^k)\|^2. \quad (18)$$

Proof The left side of (18) is

$$\begin{aligned}
\mathbb{E}\left[\|\frac{1}{m}\sum_{j\in S_{m_1}}^{m}\nabla f_j(x^k)\|^2\right] &= \frac{1}{m^2}\mathbb{E}\left[\sum_{j\in S_{m_1}}^{m}\|\nabla f_j(x^k)\|^2 + \sum_{p,q\in S_{m_1}}\sum_{p\neq q}\nabla f_p(x^k)^\top\nabla f_q(x^k)\right] \\
&= \frac{1}{mn}\sum_{j=1}^{n}\|\nabla f_j(x^k)\|^2 + \frac{m(m-1)}{n(n-1)}\cdot\frac{1}{m^2}\sum_{p\neq q}^{n}\nabla f_p^\top(x^k)\nabla f_q(x^k) \\
&= \frac{1}{mn}\left[\sum_{j=1}^{n}\|\nabla f_j(x^k)\|^2 + \frac{m-1}{n-1}\sum_{p\neq q}^{n}\nabla f_p^\top(x^k)\nabla f_q(x^k)\right] \\
&= \frac{1}{mn}\left[\frac{m-1}{n-1}\|\sum_{j=1}^{n}\nabla f_j(x^k)\|^2 + \frac{n-m}{n-1}\sum_{j=1}^{n}\|\nabla f_j(x^k)\|^2\right] \\
&= \frac{n-m}{m(n-1)}\cdot\frac{1}{n}\sum_{j=1}^{n}\|\nabla f_j(x^k)\|^2 + \frac{n(m-1)}{m(n-1)}\|\nabla f(x^k)\|^2,
\end{aligned}$$

which proves Lemma 32. ■

Now we can derive the upper bound of the noise variance of mini-batch SGD given x^k and size m.

$$\begin{aligned}
\mathbb{E}\left[\|\frac{1}{m}\sum_{j\in S_{m_1}}^{m}\nabla f_j(x^k)\|^2\right] &= \frac{n-m}{m(n-1)}\frac{1}{n}\sum_{j=1}^{n}\|\nabla f_j(x^k)\|^2 + \frac{n(m-1)}{m(n-1)}\|\nabla f(x^k)\|^2 \\
&\leq \frac{(n-m)}{m(n-1)}\frac{1}{n}\sum_{j=1}^{n}2L_j\left(f_j(x^k) - f_{j,min}\right) \\
&+ \frac{2Ln(m-1)}{m(n-1)}\left(f(x^k) - f(x^*)\right) \\
&\leq \frac{2L_{max}(n-m)}{m(n-1)}\left(f(x^k) - f(x^*) + R\right) \\
&+ \frac{2Ln(m-1)}{m(n-1)}\left(f(x^k) - f(x^*)\right) \\
&= \frac{2\left(L_{max}(n-m) + Ln(m-1)\right)}{m(n-1)}\left(f(x^k) - f(x^*)\right) \\
&+ \frac{2L_{max}R(n-m)}{m(n-1)} \\
&= C\left(f(x^k) - f(x^*)\right) + \frac{2L_{max}R(n-m)}{m(n-1)}. \quad (19)
\end{aligned}$$

The first inequality holds due to (15). $\mathbb{E}\left[2L_{S_{m_1}}\right]\mathbb{E}\left[f_{S_{m_1}}(x^k)-f_{S_{m_1},min}\right]$ is equivalent to

$$\mathbb{E}\left[2L_{S_{m_1}}\right]\mathbb{E}\left[f_{S_{m_1}}(x^k)-f_{S_{m_1},min}\right]=2\bar{L}\left(f(x^k)-f(x^*)\right)+2\bar{L}R. \tag{20}$$

Substituting (19) and (20) into (17), we obtain

$$\begin{aligned}\mathbb{E}\|g_{m,b}(x^k)\|^2 \leq & \left(\frac{C}{b}+\frac{b-1}{b}2\bar{L}\right)\left(f(x^k)-f(x^*)\right)\\ &+\left(\frac{1}{b}\cdot\frac{n-m}{m(n-1)}2L_{max}R+\frac{b-1}{b}\cdot 2\bar{L}R\right).\end{aligned} \tag{21}$$

Substituting (21) into (16), we have

$$\begin{aligned}\mathbb{E}\|x^{k+1}-x^*\|^2 &\leq (1-\mu\gamma)\|x^k-x^*\|^2-2\gamma\left(f(x^k)-f(x^*)\right)+\gamma^2\mathbb{E}\|g_{m,b}(x^k)\|^2\\ &\leq (1-\mu\gamma)\|x^k-x^*\|^2-\left(2-\left(\frac{C}{b}+\frac{b-1}{b}2\bar{L}\right)\gamma\right)\gamma\left(f(x^k)-f(x^*)\right)\\ &+\gamma^2\left(\frac{1}{b}\cdot\frac{n-m}{m(n-1)}2L_{max}R+\frac{b-1}{b}\cdot 2\bar{L}R\right)\\ &\leq (1-\mu\gamma)\|x^k-x^*\|^2+\gamma^2\left(\frac{1}{b}\cdot\frac{n-m}{m(n-1)}2L_{max}R+\frac{b-1}{b}\cdot 2\bar{L}R\right).\end{aligned}$$

The last inequality holds because stepsize $\gamma\leq 2b/\left(C+2\bar{L}(b-1)\right)$. Taking the total expectation and unrolling this recursion across T iterations, we can obtain

$$\begin{aligned}\mathbb{E}\|x^T-x^*\|^2 &\leq (1-\mu\gamma)^T\mathbb{E}\|x^0-x^*\|^2\\ &+\gamma^2\left(\frac{1}{b}\cdot\frac{2L_{max}R(n-m)}{m(n-1)}+\frac{b-1}{b}\cdot 2\bar{L}R\right)\sum_{j=0}^{T-1}(1-\mu\gamma)^j\\ &\leq (1-\mu\gamma)^T\mathbb{E}\|x^0-x^*\|^2\\ &+\gamma^2\left(\frac{1}{b}\cdot\frac{2L_{max}R(n-m)}{m(n-1)}+\frac{b-1}{b}\cdot 2\bar{L}R\right)\sum_{j=0}^{\infty}(1-\mu\gamma)^j\\ &=(1-\mu\gamma)^T\mathbb{E}\|x^0-x^*\|^2+\frac{\gamma}{\mu}\left(\frac{1}{b}\cdot\frac{2L_{max}R(n-m)}{m(n-1)}+\frac{b-1}{b}\cdot 2\bar{L}R\right).\end{aligned}$$

■

A.2 Proof of equations (4), (5) and (6)

Proof We first derive the upper bounds for per step noise variance of different methods. Equation (15) is repeatedly used in the following proof. Denote a uniformly sampled index

from $[n]$ in k-th step by i_k. The upper bound for per step noise variance of SGD given x^k is

$$\begin{aligned}\mathbb{E}\left[\|\nabla f_{i_k}(x^k)\|^2\right] = \frac{1}{n}\sum_{j=1}^{n}\|\nabla f_j(x^k)\|^2 &\leq \frac{1}{n}\sum_{j=1}^{n} 2L_j\left(f_j(x^k) - f_{j,min}\right) \\ &\leq \frac{2L_{max}}{n}\sum_{j=1}^{n}\left(f_j(x^k) - f_{j,min}\right) \\ &= 2L_{max}\left(f(x^k) - f(x^*)\right) + 2L_{max}R. \end{aligned} \tag{22}$$

We have derived the upper bound for per step noise variance of mini-batch SGD in (19). As for SGD with importance sampling, following the analysis provided by Zhao and Zhang (2015), we know that the upper bound for this noise variance is equivalent to

$$\begin{aligned}\mathbb{E}\left[\frac{1}{(nP_{i_k})^2}\|\nabla f_{i_k}(x^k)\|^2\right] &= \frac{1}{n^2}\sum_{j=1}^{n}\frac{1}{P_j}\|\nabla f_j(x^k)\|^2 \\ &\leq \frac{1}{n^2}\sum_{j=1}^{n}\frac{1}{P_j}2L_j\left(f_j(x^k) - f_{j,min}\right) \\ &= \frac{2}{n}(\sum_{j=1}^{n}L_j)\cdot\frac{1}{n}\sum_{j=1}^{n}\left(f_j(x^k) - f_{j,min}\right) \\ &= 2\bar{L}\left(f(x^k) - f(x^*)\right) + 2\bar{L}R. \end{aligned} \tag{23}$$

Equations (19), (22) and (23) share the common property since they can all be written in form of

$$A\left(f(x^k) - f(x^*)\right) + BR, \tag{24}$$

with different values of A and B. Replacing the last term in (16) by (24), we can obtain

$$\mathbb{E}\|x^{k+1} - x^*\|^2 \leq (1-\mu\gamma)\|x^k - x^*\|^2 - (2\gamma - A\gamma^2)\left(f(x^k) - f(x^*)\right) + \gamma^2 BR. \tag{25}$$

With the proper choice of stepsize γ, the second term in (25) can be absorbed since it is non-positive. Thus we can obtain

$$\mathbb{E}\|x^{k+1} - x^*\|^2 \leq (1-\mu\gamma)\|x^k - x^*\|^2 + \gamma^2 BR.$$

Unrolling these recursions, we can obtain equations (4), (5) and (6). ■

A.3 Proof of Corollary 5

Proof As long as $(1-\mu\gamma)^T\mathbb{E}\|x^0 - x^*\|^2 \leq \epsilon/2$ and $2\gamma R/b\mu D \leq \epsilon/2$, the expected optimality gap satisfies $\mathbb{E}\|x^T - x^*\|^2 \leq \epsilon$. From the definition of stepsize γ, we know that $2\gamma R/b\mu D \leq \epsilon/2$. Thus we only need to ensure that $(1-\mu\gamma)^T\mathbb{E}\|x^0 - x^*\|^2 \leq \epsilon/2$, which means

$$T\ln(1-\mu\gamma) + \ln(2\mathbb{E}\|x^0 - x^*\|^2) \leq \ln\epsilon.$$

Rearranging the terms and noticing that $\ln(1-x) \leq -x$ holds for $x \in [0,1)$, as long as we keep

$$T\gamma\mu \geq \ln\left(\frac{2\mathbb{E}\|x^0 - x^*\|^2}{\epsilon}\right),$$

the expected optimality gap is less than ϵ, which means that T satisfies

$$T \geq \max\left\{\frac{1}{2}, \frac{2b}{\mu(C + 2\bar{L}(b-1))}, \frac{4R}{\epsilon\mu^2 bD}\right\} \ln\left(\frac{2\mathbb{E}\|x^0 - x^*\|^2}{\epsilon}\right).$$

■

A.4 Proof of Theorem 6

Proof Similar to the proof of Theorem 4, we can write a decomposition for $\|x^{k+1} - x^*\|$ with a diminishing stepsize sequence,

$$\begin{aligned}
\mathbb{E}\|x^{k+1} - x^*\|^2 &\leq (1 - \mu\gamma_k)\|x^k - x^*\|^2 \\
&\quad - \left(2\gamma_k - \left(\frac{C}{b} + \frac{b-1}{b}2\bar{L}\right)\gamma_k^2\right)\left(f(x^k) - f(x^*)\right) + \gamma_k^2 \frac{2R}{bD} \\
&\leq (1 - \mu\gamma_k)\|x^k - x^*\|^2 + \gamma_k^2 \frac{2R}{bD}.
\end{aligned}$$

The last inequality holds because $\{\gamma_k\}$ is decreasing and $\gamma_0 < 2b/\left(C + 2(b-1)\bar{L}\right)$. Taking the total expectation, the rest part will be proven by induction. First, the definition of v ensures that it holds for $k = 0$. Assuming $\mathbb{E}\|x^k - x^*\|^2 \leq v/(q+k)$ holds for some $k \geq 0$, it follows that

$$\begin{aligned}
\mathbb{E}\|x^{k+1} - x^*\|^2 &\leq (1 - \mu\gamma_k)\mathbb{E}\|x^k - x^*\|^2 + \frac{2\gamma_k^2 R}{bD} \\
&\leq (1 - \frac{p\mu}{q+k})\frac{v}{q+k} + \frac{p^2}{(q+k)^2}\frac{2R}{bD} \\
&= \frac{q+k-1}{(q+k)^2}v - \frac{p\mu - 1}{(q+k)^2}v + \frac{2p^2 R}{(q+k)^2 bD} \\
&\leq \frac{q+k-1}{(q+k)^2}v \leq \frac{v}{q+k+1},
\end{aligned}$$

where the third inequality follows due to the definition of v, and the last inequality follows because $(q+k-1)(q+k+1) \leq (q+k)^2$. ■

A.5 Proof of Theorem 8

Proof Using the proof of Theorem 4, we know that

$$\mathbb{E}\|x^{k+1} - x^*\|^2 = \mathbb{E}\|x^k - x^*\|^2 - 2\gamma\langle x^k - x^*, \nabla f(x^k)\rangle + \gamma^2\mathbb{E}\|g_{m,b}(x^k)\|^2.$$

Since the objective function f is convex, we have

$$\mathbb{E}\|x^{k+1} - x^*\| \leq \|x^k - x^*\|^2 - 2\gamma\left(f(x^k) - f(x^*)\right) + \gamma^2\mathbb{E}\|g_{m,b}(x^k)\|^2. \tag{26}$$

Again using the proof of Theorem 4, the last term of (26) can be bound by

$$\mathbb{E}\|g_{m,b}(x^k)\|^2 \leq \left(\frac{C}{b} + \frac{b-1}{b}2\bar{L}\right)\left(f(x^k) - f(x^*)\right) + \frac{2R}{bD}. \tag{27}$$

Substituting (27) into (26), we have

$$\mathbb{E}\|x^{k+1} - x^*\| \leq \|x^k - x^*\|^2 - \gamma\left(2 - \frac{\left(C + (b-1)2\bar{L}\right)\gamma}{b}\right)\left(f(x^k) - f(x^*)\right) + \gamma^2\frac{2R}{bD}.$$

Rearranging the terms, summing over T iterations and taking the total expectation, we obtain

$$\begin{aligned}\gamma\left(2 - \frac{\left(C + (b-1)2\bar{L}\right)\gamma}{b}\right)\sum_{k=0}^{T-1}\mathbb{E}\left[f(x^k) - f(x^*)\right] &\leq \mathbb{E}\|x^0 - x^*\|^2 - \mathbb{E}\|x^T - x^*\|^2 + \frac{2\gamma^2 TR}{bD} \\ &\leq \mathbb{E}\|x^0 - x^*\|^2 + \frac{2\gamma^2 TR}{bD}.\end{aligned}$$

From the definition of stepsize γ, we know that

$$\frac{\gamma}{2}\sum_{k=0}^{T-1}\mathbb{E}\left[f(x^k) - f(x^*)\right] \leq \gamma\left(2 - \frac{\left(C + (b-1)2\bar{L}\right)\gamma}{b}\right)\sum_{k=0}^{T-1}\mathbb{E}\left[f(x^k) - f(x^*)\right],$$

thus we have

$$\frac{\gamma}{2}\sum_{k=0}^{T-1}\mathbb{E}\left[f(x^k) - f(x^*)\right] \leq \mathbb{E}\|x^0 - x^*\|^2 + \frac{2\gamma^2 TR}{bD}.$$

Dividing $\gamma T/2$ on the both side, due to the convexity of the objective function f, we have

$$\mathbb{E}\left[f(\hat{x}) - f(x^*)\right] \leq \frac{2\mathbb{E}\|x^0 - x^*\|^2}{T\gamma} + \frac{4\gamma R}{bD}.$$

■

A.6 Proof of Corollary 9

Proof From the definition of the stepsize γ, we can verify that $4\gamma R/bD \leq \epsilon/2$. To achieve ϵ-optimality, we need to make sure that the first term in (7) is not greater than $\epsilon/2$. That is

$$\frac{2\mathbb{E}\|x^0 - x^*\|^2}{T\gamma} \leq \frac{\epsilon}{2}.$$

Rearranging the terms, we know that T should satisfy

$$T \geq \frac{4\mathbb{E}\|x^0 - x^*\|^2}{\epsilon\min\{3b/2(C + (b-1)2\bar{L}), \epsilon bD/8R\}}.$$

■

A.7 Proof of Theorem 10

Proof From the L-smoothness of the objective function f, we have

$$f(x^{k+1}) \leq f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{L}{2}\|x^{k+1} - x^k\|^2.$$

Take the conditional expectation on x^k,

$$\mathbb{E}[f(x^{k+1})] - f(x^k) \leq -\gamma \|\nabla f(x^k)\|^2 + \frac{1}{2}\gamma^2 L \mathbb{E}\|g_{m,b}(x^k)\|^2. \tag{28}$$

From (17), we can bound $\mathbb{E}\|g_{m,b}(x^k)\|^2$ by,

$$\mathbb{E}\|g_{m,b}(x^k)\|^2 \leq \frac{1}{b}\mathbb{E}\|\nabla f_{S_{m_1}}(x^k)\|^2 + \frac{b-1}{b} 2\bar{L}\left(f(x^k) - f_{min}\right). \tag{29}$$

Substituting (29) into (28), we obtain

$$\begin{aligned}
\mathbb{E}[f(x^{k+1})] - f(x^k) &\leq -\gamma \|\nabla f(x^k)\|^2 + \frac{\gamma^2 L}{2}\left(\frac{1}{b}\mathbb{E}\|\nabla f_{S_{m_1}}(x^k)\|^2 + \frac{b-1}{b} 2\bar{L}\left(f(x^k) - f_{min}\right)\right) \\
&= \gamma\left(\frac{\gamma L}{2b} \cdot \frac{n(m-1)}{m(n-1)} - 1\right)\|\nabla f(x^k)\|^2 \\
&+ \frac{\gamma^2 L}{2b}\left(\frac{n-m}{m(n-1)}\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x^k)\|^2 + (b-1)2\bar{L}\left(f(x^k) - f_{min}\right)\right) \\
&\leq \gamma\left(\frac{\gamma L}{2b} \cdot \frac{n(m-1)}{m(n-1)} - 1\right)\|\nabla f(x^k)\|^2 + \frac{\gamma^2 L}{bD}\left(f(x^k) - f_{min}\right).
\end{aligned}$$

The last inequality holds due to (15). Substracting f_{min} on the both side, we obtain

$$\mathbb{E}[f(x^{k+1})] - f_{min} \leq \gamma\left(\frac{\gamma L}{2b} \cdot \frac{n(m-1)}{m(n-1)} - 1\right)\|\nabla f(x^k)\|^2 + \left(1 + \frac{\gamma^2 L}{bD}\right)\left(f(x^k) - f_{min}\right).$$

Since $\gamma \leq (2b-1)/L$ and $m(n-1)/n(m-1) \leq 1$, we have

$$\mathbb{E}[f(x^{k+1})] - f_{min} \leq -\frac{\gamma}{2b}\|\nabla f(x^k)\|^2 + \left(1 + \frac{\gamma^2 L}{bD}\right)\left(f(x^k) - f_{min}\right).$$

Let $\delta_k = \mathbb{E}\left[f(x^k)\right] - f_{min}$. Taking total expectation and rearranging the terms, we have

$$\frac{\gamma}{2b}\left(1 + \frac{\gamma^2 L}{bD}\right)^{-1}\mathbb{E}\|\nabla f(x^k)\|^2 \leq \delta_k - \left(1 + \frac{\gamma^2 L}{bD}\right)^{-1}\delta_{k+1}. \tag{30}$$

Consider the sequence $\{\alpha_k\}_{k=0}$, where $\alpha_k = (1 + \gamma^2 L/bD)^{-k}\alpha_0$ and $\alpha_0 > 0$ is a constant. Multiplying α_k on the both side of (30), we obtain

$$\frac{\gamma}{2b}\left(1 + \frac{\gamma^2 L}{bD}\right)^{-1}\alpha_k\mathbb{E}\|\nabla f(x^k)\|^2 \leq \alpha_k\delta_k - \alpha_{k+1}\delta_{k+1}.$$

Summing over T iterations and rearranging the terms, we obtain

$$\frac{\gamma}{2b}\left(1+\frac{\gamma^2 L}{bD}\right)^{-1}\sum_{k=0}^{T-1}\alpha_k\mathbb{E}\|\nabla f(x^k)\|^2 \leq \alpha_0\delta_0 - \alpha_T\delta_T \leq \alpha_0\delta_0.$$

since the series $\sum_{k=0}^{T-1}\alpha_k$ is finite, we have

$$\frac{\gamma}{2b}\left(1+\frac{\gamma^2 L}{bD}\right)^{-1}\min_{k=0,\ldots,T-1}\mathbb{E}\|\nabla f(x^k)\|^2 \leq \frac{\alpha_0\delta_0}{\sum_{k=0}^{T-1}\alpha_k}.$$

Rearranging the terms, we can conclude

$$\begin{aligned}
\min_{k=0,\ldots,T-1}\mathbb{E}\|\nabla f(x^k)\|^2 &\leq \frac{2b}{\gamma}\left(1+\frac{\gamma^2 L}{bD}\right)\alpha_0\delta_0\left(\frac{1}{\alpha_0}\frac{1-\left(1+\frac{\gamma^2 L}{bD}\right)^{-1}}{1-\left(1+\frac{\gamma^2 L}{bD}\right)^{-T}}\right)\\
&\leq \frac{2b}{\gamma}\left(\frac{\frac{\gamma^2 L}{bD}}{1-\left(1+\frac{\gamma^2 L}{bD}\right)^{-T}}\right)\delta_0\\
&\leq \frac{2\gamma L}{D}\left(\frac{1}{1-\left(1+\frac{\gamma^2 L}{bD}\right)^{-T}}\right)\delta_0\\
&\leq \frac{2\gamma L}{D}\left(\frac{1+\left(\frac{\gamma^2 L}{bD}\right)T}{\left(\frac{\gamma^2 L}{bD}\right)T}\right)\delta_0\\
&= \frac{2\gamma L}{D}\left(1+\frac{bD}{\gamma^2 LT}\right)\delta_0.
\end{aligned}$$

The last inequality holds because $(1+\beta)^n \geq 1+n\beta$ for $n\in\mathbb{N}$, $n\geq 1$ and $\beta > -1$. ■

A.8 Proof of Corollary 11

Proof To achieve ϵ-optimality, T should satisfy

$$\frac{2\gamma L}{D}\left(1+\frac{bD}{\gamma^2 LT}\right)\delta_0 \leq \epsilon,$$

which is equivalent to

$$\frac{bD}{\gamma LT}+\gamma \leq \frac{\epsilon D}{2L\delta_0}.$$

Since $\gamma \leq \epsilon D/4L\delta_0$. As long as inequality

$$\frac{bD}{\gamma LT}+\gamma \leq \frac{bD}{\gamma LT}+\frac{\epsilon D}{4L\delta_0} \leq \frac{\epsilon D}{2L\delta_0}$$

holds, we can achieve ϵ-optimality. Rearranging the terms, we can conclude

$$T \geq \frac{4\delta_0 b}{\epsilon\gamma} = \frac{4\delta_0 b}{\epsilon} \max\left\{\frac{L}{2b-1}, \frac{4L\delta_0}{\epsilon D}\right\}.$$

■

A.9 Proof of Theorem 12

Proof Let $D_m^i = \{S \mid S \subset D_m, i \in S\}$. Following the proof of Theorem 4, we first prove that the grafting gradient using sampling without replacement is an unbiased estimator with respect to the full gradient.

$$\begin{aligned}
\mathbb{E}\frac{1}{bP_{S_{m_{r_1}}}}\frac{\partial f_{S_{m_{r_1}}}}{\partial x_1}\left(x^k\right) &= \mathbb{E}\left[\mathbb{E}\left[\frac{1}{bP_{S_{m_{r_1}}}}\frac{\partial f_{S_{m_{r_1}}}}{\partial x_1}\left(x^k\right) \mid S_m^b\right]\right] \\
&= \mathbb{E}\left[\frac{1}{b}\sum_{i=1}^{b}\frac{1}{P_{S_{m_i}}}\frac{\partial f_{S_{m_i}}}{\partial x_1}\left(x^k\right)\cdot P_{S_{m_i}}\right] = \mathbb{E}\left[\frac{1}{b}\sum_{i=1}^{b}\frac{\partial f_{S_{m_i}}}{\partial x_1}\left(x^k\right)\right] \\
&= \mathbb{E}\left[\frac{1}{bm}\sum_{i=1}^{n}\frac{\partial f_i}{\partial x_1}(x^k)\cdot\left(\sum_{S\in D_m^i}\mathbb{I}_{\{S\in S_m^b\}}\right)\right] \\
&= \frac{1}{bm}\sum_{i=1}^{n}\frac{\partial f_i}{\partial x_1}\left(x^k\right)\cdot \mathrm{C}_{n-1}^{m-1}\cdot\frac{b}{\mathrm{C}_n^m} = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial f_i}{\partial x_1}(x^k),
\end{aligned}$$

which proves that the grafting gradient using sampling without replacement is an unbiased estimator with respect to the full gradient. Recall the recursion we wrote the proof of Theorem 4.

$$\mathbb{E}\|x^{k+1}-x^*\|^2 \leq (1-\mu\gamma)\|x^k-x^*\|^2 - 2\gamma\left(f(x^k)-f(x^*)\right) + \gamma^2\mathbb{E}\|g_{m,b}(x^k)\|^2. \tag{31}$$

Since f_i is L_i-smooth, the theoretical bound for $\mathbb{E}\|g_{m,k}(x^k)\|^2$ can be derived as follows.

$$\begin{aligned}
\mathbb{E}\|g_{m,b}(x^k)\|^2 &= \mathbb{E}\left[\frac{1}{b^2}\sum_{i=1}^{b}\frac{1}{P_{S_{m_i}}}\|\nabla f_{S_{m_i}}(x^k)\|^2\right] \\
&\leq \mathbb{E}\left[\frac{1}{b^2}\sum_{i=1}^{b}\frac{1}{P_{S_{m_i}}}2L_{S_{m_i}}\left(f_{S_{m_i}}(x^k)-f_{S_{m_i},min}\right)\right] \\
&= \mathbb{E}\left[\frac{1}{b^2}\left(\sum_{i=1}^{b}2L_{S_{m_i}}\right)\left(\sum_{i=1}^{b}f_{S_{m_i}}-f_{S_{m_i},min}\right)\right]
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{b^2}\mathbb{E}\Bigg[\frac{2}{m}\left(\sum_{i=1}^{n} L_i \cdot \left(\sum_{S\in D_m^i} \mathbb{I}_{\{S\in S_m^b\}}\right)\right) \\
&\cdot \frac{1}{m}\left(\sum_{i=1}^{n}\Big(f_i(x^k) - f_{i,min}\Big)\left(\sum_{S\in D_m^i} \mathbb{I}_{\{S\in S_m^b\}}\right)\right)\Bigg] \\
&= \frac{1}{b^2m^2}\mathbb{E}\left[\sum_{i=1}^{n} 2L_i\Big(f_i(x^k) - f_{i,min}\Big)\left(\sum_{S\in D_m^i} \mathbb{I}_{\{S\in S_m^b\}}\right)^2\right] \\
&+ \frac{1}{b^2m^2}\mathbb{E}\left[\sum_{p\neq q}^{n} 2L_p\Big(f_q(x^k) - f_{q,min}\Big)\left(\sum_{U\in D_m^p} \mathbb{I}_{\{U\in S_m^b\}}\right)\left(\sum_{V\in D_m^q} \mathbb{I}_{\{V\in S_m^b\}}\right)\right]. \qquad (32)
\end{aligned}
$$

Now we first deal with the term $\mathbb{E}\left[\sum_{S\in D_m^i} \mathbb{I}_{\{S\in S_m^b\}}\right]^2$.

$$
\begin{aligned}
\mathbb{E}\left[\left(\sum_{S\in D_m^i} \mathbb{I}_{\{S\in S_m^b\}}\right)^2\right] &= \mathbb{E}\left(\sum_{S\in D_m^i} \mathbb{I}_{\{S\in S_m^b\}} + \sum_{\substack{U,V\in D_m^i \\ U\neq V}} \mathbb{I}_{\{U\in S_m^b\}}\cdot\mathbb{I}_{\{V\in S_m^b\}}\right) \\
&= \mathrm{C}_{n-1}^{m-1}\frac{b}{\mathrm{C}_n^m} + \mathrm{C}_{n-1}^{m-1}(\mathrm{C}_{n-1}^{m-1}-1)\frac{b(b-1)}{\mathrm{C}_n^m(\mathrm{C}_n^m-1)} \\
&= \frac{mb}{n}\left(1 + \frac{(\mathrm{C}_{n-1}^{m-1}-1)(b-1)}{\mathrm{C}_n^m - 1}\right) = M_1. \qquad (33)
\end{aligned}
$$

Denote $D_m^{p,q} = \{S \mid S \subset D_m \text{ and } p, q \in S\}$. Likewise, the term $\mathbb{E}\left(\sum_{U\in D_m^p} \mathbb{I}_{\{U\in S_m^b\}}\right)\left(\sum_{V\in D_m^q} \mathbb{I}_{\{V\in S_m^b\}}\right)$ is equivalent to

$$
\begin{aligned}
&\mathbb{E}\left(\sum_{U,V\in D_m^{p,q}} \mathbb{I}_{\{U\in S_m^b\}}\right) + \mathbb{E}\left(\sum_{\substack{U\in D_m^p, V\in D_m^q \\ V,U\notin D_m^{p,q}}} \mathbb{I}_{\{U\in S_m^b\}}\cdot\mathbb{I}_{\{V\in S_m^b\}}\right) \\
&= \mathrm{C}_{n-2}^{m-2}\cdot\frac{b}{\mathrm{C}_n^m} + \Big(\big(\mathrm{C}_{n-1}^{m-1}\big)^2 - \mathrm{C}_{n-2}^{m-2}\Big)\cdot\frac{b(b-1)}{\mathrm{C}_n^m(\mathrm{C}_n^m-1)} \\
&= \frac{m(m-1)b}{n(n-1)} + \frac{mb(b-1)\mathrm{C}_{n-1}^{m-1}}{n(\mathrm{C}_n^m-1)} - \frac{m(m-1)b(b-1)}{n(n-1)(\mathrm{C}_n^m-1)} \\
&= \frac{mb(b-1)\mathrm{C}_{n-1}^{m-1}}{(\mathrm{C}_n^m-1)} + \frac{m(m-1)b}{n(n-1)}\left(1 - \frac{b-1}{\mathrm{C}_n^m-1}\right) = M_2. \qquad (34)
\end{aligned}
$$

Combining (32), (33) and (34), we have

$$\begin{aligned}
\mathbb{E}\|g_{m,b}(x^k)\|^2 &\leq \frac{2}{b^2m^2}\left[M_1\cdot\sum_{i=1}^{n}L_i\left(f_i(x^k)-f_{i,min}\right)+M_2\cdot\sum_{p\neq q}^{n}L_p\left(f_q(x^k)-f_{q,min}\right)\right]\\
&=\frac{2}{b^2m^2}\left[M_2\cdot\left(\sum_{i=1}^{n}L_i\right)\left(\sum_{i=1}^{n}f_i(x^k)-f_{i,min}\right)\right]\\
&+\frac{2}{b^2m^2}\left[(M_1-M_2)\cdot\sum_{i=1}^{n}L_i\left(f_i(x^k)-f_{i,min}\right)\right]\\
&=\frac{n^2}{b^2m^2}\cdot M_2\cdot 2\bar{L}\left(f(x^k)-f_{min}\right)\\
&+\frac{2n}{b^2m^2}(M_1-M_2)\cdot\frac{1}{n}\sum_{i=1}^{n}L_i\left(f_i(x^k)-f_{i,min}\right)\\
&\leq\frac{n^2}{b^2m^2}\cdot M_2\cdot 2\bar{L}\left(f(x^k)-f_{min}\right)\\
&+\frac{n}{b^2m^2}(M_1-M_2)\cdot 2\tilde{L}\left(f(x^k)-f_{min}\right). \qquad (35)
\end{aligned}$$

Substituting (35) into (31), we can obtain

$$\begin{aligned}
\mathbb{E}\|x^{k+1}-x^*\|^2 &\leq (1-\mu\gamma)\|x^k-x^*\|^2-2\gamma\left(f(x^k)-f(x^*)\right)\\
&+\gamma^2\frac{n^2}{b^2m^2}\cdot M_2\cdot 2\bar{L}\left(f(x^k)-f_{min}\right)\\
&+\gamma^2\frac{n}{b^2m^2}(M_1-M_2)\cdot 2\tilde{L}\left(f(x^k)-f_{min}\right)\\
&\leq(1-\mu\gamma)\|x^k-x^*\|^2\\
&-2\gamma\left(1-\left(\frac{n^2\bar{L}}{b^2m^2}M_2+\frac{n\tilde{L}}{b^2m^2}(M_1-M_2)\right)\gamma\right)\left(f(x^k)-f(x^*)\right)\\
&+\frac{2\gamma^2R}{b^2m^2}\left(n^2\cdot M_2\cdot\bar{L}+n(M_1-M_2)\cdot\tilde{L}\right). \qquad (36)
\end{aligned}$$

From the definition of stepsize γ, we know that the second term in the last inequality of (36) can be absorbed. Thus we can derive

$$\mathbb{E}\|x^{k+1}-x^*\|^2\leq(1-\mu\gamma)\|x^k-x^*\|^2+\frac{2\gamma^2R}{b^2m^2}\left(n^2\cdot M_2\cdot\bar{L}+n(M_1-M_2)\cdot\tilde{L}\right).$$

Taking total expectation and unrolling this recursion across T iterations, we can obtain

$$\mathbb{E}\|x^T-x^*\|^2\leq(1-\mu\gamma)^T\mathbb{E}\|x^0-x^*\|^2+\frac{2\gamma RM}{\mu b^2m^2}.$$

■

A.10 Proof of Corollary 16

Proof Following the proof of Corollary 5, we know that as long as $(1-\mu\gamma)^T\mathbb{E}\|x^0-x^*\|^2 \leq \epsilon/2$. GGD-WoR can achieve ϵ-optimality under strongly-convex assumption, which means

$$T\ln(1-\mu\gamma)+\ln\left(2\mathbb{E}\|x^0-x^*\|^2\right)\leq \ln\epsilon.$$

Rearranging the terms and noticing that $\ln(1-x)\leq -x$ holds for $x\in[0,1)$, as long as we keep

$$T\gamma\mu \geq \ln\left(\frac{2\mathbb{E}\|x^0-x^*\|^2}{\epsilon}\right),$$

the expected optimality gap is less than ϵ, which means that T should satisfy

$$T\geq \max\left\{\frac{1}{2},\frac{M}{b^2m^2\mu},\frac{4RM}{\epsilon\mu^2b^2m^2}\right\}\ln\left(\frac{2\mathbb{E}\|x^0-x^*\|^2}{\epsilon}\right).$$

■

A.11 Proof of Theorem 17

Proof Using the proof of Theorem 8, we have

$$\mathbb{E}\|x^{k+1}-x^*\|^2 \leq \|x^k-x^*\|^2-2\gamma\left(f(x^k)-f(x^*)\right)+\gamma^2\mathbb{E}\|g_{m,b}(x^k)\|^2. \tag{37}$$

From (35), we know that $\mathbb{E}\|g_{m,b}(x^k)\|^2$ can be bounded by

$$\mathbb{E}\|g_{m,b}(x^k)\|^2 \leq \frac{2M}{b^2m^2}\left(f(x^k)-f(x^*)\right)+\frac{2RM}{b^2m^2}. \tag{38}$$

Combining (37) and (38), we can derive

$$\mathbb{E}\|x^{k+1}-x^*\|^2 \leq \|x^k-x^*\|^2-2\gamma\left(1-\frac{\gamma M}{b^2m^2}\right)\left(f(x^k)-f(x^*)\right)+\frac{2\gamma^2RM}{b^2m^2}.$$

Rearranging the terms and noting $\gamma\leq b^2m^2/2M$, we can obtain

$$\gamma\left(f(x^k)-f(x^*)\right)\leq \|x^k-x^*\|^2-\mathbb{E}\|x^{k+1}-x^*\|^2+\frac{2\gamma^2RM}{b^2m^2}.$$

Taking total expectation and summing over T iterations, we have

$$\gamma\sum_{k=0}^{T-1}\mathbb{E}\left[f(x^k)-f(x^*)\right]\leq \mathbb{E}\|x^0-x^*\|^2+\frac{2\gamma^2RMT}{b^2m^2}.$$

Dividing γT on both side, due to the convexity of the objective function f, we have

$$\mathbb{E}\left[f(\hat{x})-f(x^*)\right]\leq \frac{\mathbb{E}\|x^0-x^*\|^2}{T\gamma}+\frac{2\gamma RM}{b^2m^2}.$$

■

A.12 Proof of Corollary 18

Proof From the definition of stepsize γ, we know that as long as $\mathbb{E}\|x^0 - x^*\|^2/T\gamma \leq \epsilon/2$, that is

$$T \geq \frac{2\mathbb{E}\|x^0 - x^*\|^2}{\epsilon\gamma},$$

holds, GGD-WoR can achieve ϵ-optimality. since $\gamma = b^2m^2/2M \cdot \min\{1, \epsilon/2R\}$, we can obtain

$$T \geq \frac{4M\mathbb{E}\|x^0 - x^*\|^2}{\epsilon b^2 m^2 \min\{1, \epsilon/2R\}}.$$

■

A.13 Proof of Theorem 19

Proof Using the proof of Theorem 10, we can obtain

$$\mathbb{E}\left[f(x^{k+1})\right] - f(x^k) \leq -\gamma\|\nabla f(x^k)\|^2 + \frac{\gamma^2 L}{2}\mathbb{E}\|g_{m,b}(x^k)\|^2. \tag{39}$$

Substituting (38) into (39), we have

$$\mathbb{E}\left[f(x^{k+1})\right] - f(x^k) \leq -\gamma\|\nabla f(x^k)\|^2 + \frac{\gamma^2 LM}{b^2m^2}\left(f(x^k) - f_{min}\right).$$

Consider the sequence $\{\alpha_k\}$, where $\alpha_k = (1 + \gamma^2 LM/b^2m^2)\alpha_0$ and $\alpha_0 > 0$ is a constant. Recall the technical tricks used in proof of Theorem 10, we can proceed the proof with

$$\gamma\left(1 + \frac{\gamma^2 LM}{b^2m^2}\right)^{-1}\alpha_k\mathbb{E}\|\nabla f(x^k)\|^2 \leq \alpha_k\delta_k - \alpha_{k+1}\delta_{k+1}.$$

Summing over T iterations and rearranging the terms, we have

$$\gamma\left(1 + \frac{\gamma^2 LM}{b^2m^2}\right)^{-1}\min_{k=0,\ldots,T-1}\mathbb{E}\|\nabla f(x^k)\|^2 \leq \frac{\alpha_0\delta_0}{\sum_{k=0}^{T-1}\alpha_k}.$$

Rearranging the terms, we can conclude

$$\begin{aligned}
\min_{k=0,\ldots,T-1}\mathbb{E}\|\nabla f(x^k)\|^2 &\leq \frac{1}{\gamma}\left(1 + \frac{\gamma^2 LM}{b^2m^2}\right)\frac{\alpha_0\delta_0}{\sum_{k=0}^{T-1}\alpha_k} \\
&\leq \frac{\gamma LM}{b^2m^2}\left(\frac{1}{1 - \left(1 + \frac{\gamma^2 LM}{b^2m^2}\right)^{-T}}\right)\delta_0 \\
&\leq \frac{\gamma LM}{b^2m^2}\left(1 + \frac{b^2m^2}{\gamma^2 LMT}\right)\delta_0 \\
&\leq \frac{\epsilon}{2} + \frac{\delta_0}{\gamma T}.
\end{aligned}$$

■

Appendix B. Proofs of Theorems and Corollaries in Section 5

In this section, we prove the theorems and corollaries in Section 5.

B.1 Proof of Theorem 21

Proof Recall that the decomposition used in the proof of Theorem 4,

$$\|x_s^{k+1} - x^*\|^2 = \|x_s^k - \gamma \tilde{g}_{m,b}^k - x^*\|^2 = \|x_s^k - x^*\|^2 - 2\gamma\langle x_s^k - x^*, \tilde{g}_{m,b}^k\rangle + \gamma^2\|\tilde{g}_{m,b}^k\|^2.$$

Taking the conditional expectation on x_s^k and all past, we can obtain

$$\mathbb{E}\|x_s^{k+1} - x^*\|^2 = \|x_s^k - x^*\|^2 - 2\gamma\langle x_s^k - x^*, \nabla f(x_s^k)\rangle + \gamma^2\mathbb{E}\|\tilde{g}_{m,b}^k\|^2. \tag{40}$$

Now we deal with the third term in (40),

$$\mathbb{E}\|\tilde{g}_{m,b}^k\|^2 = \mathbb{E}\|g_{m,b}(x_s^k) - g_{m,b}(\bar{x}) + \nabla f(\bar{x})\|^2 \leq 2\mathbb{E}\|g_{m,b}(x_s^k) - g_{m,b}(\bar{x})\|^2 + 2\|\nabla f(\bar{x})\|^2. \tag{41}$$

We first deal with $\mathbb{E}\|g_{m,b}(x_s^k) - g_{m,b}(\bar{x})\|^2$, similar to the proof of Theorem 4,

$$\begin{aligned}
\mathbb{E}\left\|g_{m,b}(x_s^k) - g_{m,b}\left(\bar{x}\right)\right\|^2 &= \mathbb{E}\sum_{i=1}^{d}\frac{1}{b^2 P_{S_{mr_i}}^2}\left(\frac{\partial f_{S_{mr_i}}}{\partial x_i}(x_s^k) - \frac{\partial f_{S_{mr_i}}}{\partial x_i}\left(\bar{x}\right)\right)^2 \\
&= \sum_{i=1}^{d}\mathbb{E}\left[\mathbb{E}\left[\frac{1}{b^2 P_{S_{mr_i}}^2}\left(\frac{\partial f_{S_{mr_i}}}{\partial x_i}(x_s^k) - \frac{\partial f_{S_{mr_i}}}{\partial x_i}\left(\bar{x}\right)\right)^2 \mid S_m^b\right]\right] \\
&= \sum_{i=1}^{d}\mathbb{E}\left[\frac{1}{b^2}\sum_{j=1}^{b}\frac{1}{P_{S_{m_j}}}\left(\frac{\partial f_{S_{m_j}}}{\partial x_i}(x_s^k) - \frac{\partial f_{S_{m_j}}}{\partial x_i}\left(\bar{x}\right)\right)^2\right] \\
&= \mathbb{E}\left[\frac{1}{b^2}\sum_{j=1}^{b}\frac{1}{P_{S_{m_j}}}\|\nabla f_{S_{m_j}}(x_s^k) - \nabla f_{S_{m_j}}(\bar{x})\|^2\right].
\end{aligned}$$

Recalling the resampling probability defined in Algorithm 2, we have

$$\begin{aligned}
\mathbb{E}\|g_{m,b}(x^k) - g_{m,b}(\bar{x})\|^2 &= \mathbb{E}\left[\frac{1}{b^2}\left(\sum_{j=1}^{b}\|\nabla f_{S_{m_j}}(x_s^k) - \nabla f_{S_{m_j}}(\bar{x})\|\right)^2\right] \\
&= \frac{1}{b^2}\mathbb{E}\left[\sum_{j=1}^{b}\|\nabla f_{S_{m_j}}(x_s^k) - \nabla f_{S_{m_j}}(\bar{x})\|^2\right] \\
&+ \frac{1}{b^2}\mathbb{E}\left[\sum_{u\neq t}^{b}\|\nabla f_{S_{m_u}}(x_s^k) - \nabla f_{S_{m_u}}(\bar{x})\| \cdot \|\nabla f_{S_{m_t}}(x_s^k) - \nabla f_{S_{m_t}}(\bar{x})\|\right] \\
&= \frac{1}{b}\mathbb{E}\|\nabla f_{S_{m_1}}(x_s^k) - \nabla f_{S_{m_1}}(\bar{x})\|^2 \\
&+ \frac{b-1}{b}\left(\mathbb{E}\|\nabla f_{S_{m_1}}(x_s^k) - \nabla f_{S_{m_1}}(\bar{x})\|\right)^2.
\end{aligned} \tag{42}$$

We first deal with the second term in the last equality of (42).

$$\begin{aligned}\left(\mathbb{E}\|\nabla f_{S_{m_1}}(x_s^k)-\nabla f_{S_{m_1}}(\bar{x})\|\right)^2 &\le \mathbb{E}\left[\frac{1}{m}\sum_{i\in S_{m_1}}\|\nabla f_i(x_s^k)-\nabla f_i(\bar{x})\|\right]^2\\ &\le \left(\frac{1}{n}\sum_{i=1}^n\|\nabla f_i(x_s^k)-\nabla f_i(\bar{x})\|\right)^2\\ &\le \frac{1}{n}\sum_{i=1}^n\|\nabla f_i(x_s^k)-\nabla f_i(\bar{x})\|^2, \end{aligned}\tag{43}$$

where the last inequality holds due to Hölder inequality. The last inequality can be further bounded by

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^n\|\nabla f_i(x_s^k)-\nabla f_i(\bar{x})\|^2 &= \frac{1}{n}\sum_{i=1}^n\|\nabla f_i(x_s^k)-\nabla f_i(x^*)+\nabla f_i(x^*)-\nabla f_i(\bar{x})\|^2\\ &\le \frac{2}{n}\sum_{i=1}^n\|\nabla f_i(x_s^k)-\nabla f_i(x^*)\|^2+\frac{2}{n}\sum_{i=1}^n\|\nabla f_i(x^*)-f_i(\bar{x})\|^2\\ &\le \frac{4L}{n}\sum_{i=1}^n\left(f_i(x_s^k)-f_i(x^*)-\langle\nabla f_i(x^*),x_s^k-x^*\rangle\right)\\ &+\frac{4L}{n}\sum_{i=1}^n\left(f_i(\bar{x})-f_i(x^*)-\langle\nabla f_i(x^*),\bar{x}-x^*\rangle\right)\\ &= 4L\left(f(x_s^k)-f(x^*)\right)+4L\left(f(\bar{x})-f(x^*)\right). \end{aligned}\tag{44}$$

The last inequality holds because of (14). Now we back to the first term in the last equality of (42). From (19), if we replace $\nabla f_i(x^k)$ with $\nabla f_i(x_s^k)-\nabla f_i(\bar{x})$, then we have

$$\begin{aligned}\mathbb{E}\|\nabla f_{S_{m_1}}(x_s^k)-\nabla f_{S_{m_1}}(\bar{x})\|^2 &= \mathbb{E}\left[\|\frac{1}{m}\sum_{i\in S_{m_1}}\left(\nabla f_i(x_s^k)-\nabla f_i(\bar{x})\right)\|^2\right]\\ &= \frac{n(m-1)}{m(n-1)}\|\nabla f(x_s^k)-\nabla f(\bar{x})\|^2\\ &+\frac{n-m}{m(n-1)}\frac{1}{n}\sum_{i=1}^n\|\nabla f_i(x_s^k)-\nabla f_i(\bar{x})\|^2. \end{aligned}\tag{45}$$

Substituting (43) and (45) into (42), we have

$$\begin{aligned}\mathbb{E}\|g_{m,b}(x_s^k)-g_{m,b}(\bar{x})\|^2 &\le \frac{n(m-1)}{bm(n-1)}\|\nabla f(x_s^k)-\nabla f(\bar{x})\|^2\\ &+\left(\frac{n-m}{bm(n-1)}+\frac{b-1}{b}\right)\frac{1}{n}\sum_{i=1}^n\|\nabla f_i(x_s^k)-\nabla f_i(\bar{x})\|^2. \end{aligned}$$

From L-smoothness and (44), we can obtain

$$\mathbb{E}\|g_{m,b}(x_s^k) - g_{m,b}(\bar{x})\|^2 \leq 4L\left(f(x_s^k) - f(x^*)\right) + 4L\left(f(\bar{x}) - f(x^*)\right). \tag{46}$$

Substituting (46) into (41), we have

$$\begin{aligned}\mathbb{E}\|\tilde{g}_{m,b}^k\|^2 &\leq 2\mathbb{E}\|g_{m,b}(x_s^k) - g_{m,b}(\bar{x})\|^2 + 2\|\nabla f(\bar{x})\|^2 \\ &\leq 8L\left(f(x_s^k) - f(x^*)\right) + 12L\left(f(\bar{x}) - f(x^*)\right).\end{aligned} \tag{47}$$

Substituting (47) into (40), we can obtain

$$\begin{aligned}\mathbb{E}\|x_s^{k+1} - x^*\|^2 &\leq \|x_s^k - x^*\|^2 - 2\gamma\langle x_s^k - x^*, \nabla f(x_s^k)\rangle + 8L\gamma^2\left(f(x_s^k) - f(x^*)\right) \\ &\quad + 12L\gamma^2\left(f(\bar{x}) - f(x^*)\right) \\ &\leq (1-\gamma\mu)\|x_s^k - x^*\|^2 - 2\gamma(1-4\gamma L)\left(f(x_s^k) - f(x^*)\right) \\ &\quad + 12L\gamma^2\left(f(\bar{x}) - f(x^*)\right).\end{aligned} \tag{48}$$

The second inequality holds due to μ-strong convexity. Using Lemma 31, we know that $(1-\gamma\mu) > 0$ since $\gamma \leq 1/16L$ and $L \geq \mu$. Taking the total expectation and iterating over $k = 0, 1, ..., q-1$, we have

$$\begin{aligned}\mathbb{E}\|x_s^q - x^*\|^2 &\leq (1-\gamma\mu)^q\mathbb{E}\|x_s^0 - x^*\|^2 \\ &\quad - 2\gamma(1-4L\gamma)\sum_{k=0}^{q-1}(1-\gamma\mu)^{q-1-k}\mathbb{E}\left[f(x_s^k) - f(x^*)\right] \\ &\quad + 12L\gamma^2\mathbb{E}\left[f(\bar{x}) - f(x^*)\right]\sum_{k=0}^{q-1}(1-\gamma\mu)^{q-1-k}.\end{aligned}$$

Considering the option (a) in the outer loop and the definitions of V_q and p_k in (10), we have

$$\begin{aligned}\mathbb{E}\|x_s^q - x^*\|^2 &\leq (1-\gamma\mu)^q\mathbb{E}\|x_{s-1}^q - x^*\|^2 - 2\gamma(1-4L\gamma)V_q\sum_{k=0}^{q-1}p_k\mathbb{E}\left[f(x_s^k) - f(x^*)\right] \\ &\quad + 12L\gamma^2V_q\mathbb{E}\left[f(\bar{x}) - f(x^*)\right].\end{aligned} \tag{49}$$

Define Lyapunov function Φ_s as follows.

$$\Phi_s = \|x_s^q - x^*\|^2 + \Psi_s, \text{ where } \Psi_s = 24L\gamma^2V_q\mathbb{E}\left[f(\bar{x}_s) - f(x^*)\right].$$

Noticing that $1-\gamma\mu > 0$ implies that $p_k > 0$ for $k = 0, ..., q-1$ and $\sum_{k=0}^{q-1}p_k = 1$, we have

$$f(\bar{x}_s) - f(x^*) = f\left(\sum_{k=0}^{q-1}p_kx_s^k\right) - f(x^*) \leq \sum_{k=0}^{q-1}p_k\left(f(x_s^k) - f(x^*)\right).$$

The last inequality holds using Jensen's inequality and the fact that f is convex. Hence, the expectation of Ψ_s can be bounded by

$$\mathbb{E}[\Psi_s] \leq 24L\gamma^2 V_q \sum_{k=0}^{q-1} p_k \mathbb{E}\left[f(x_s^k) - f(x^*)\right]. \tag{50}$$

Taking the total expectation of Lyapunov function, we have

$$\mathbb{E}[\Phi_s] = \mathbb{E}\|x_s^k - x^*\|^2 + \mathbb{E}[\Psi_s]. \tag{51}$$

Substituting (49) into (51), we have

$$\begin{aligned}\mathbb{E}[\Phi_s] \leq{}& (1-\gamma\mu)^q \mathbb{E}\|x_{s-1}^q - x^*\|^2 - 2\gamma(1-4L\gamma)V_q \sum_{k=0}^{q-1} p_k \mathbb{E}\left[f(x_s^k) - f(x^*)\right] \\ &+ 12L\gamma^2 V_q \mathbb{E}\left[f(\bar{x}) - f(x^*)\right] + \mathbb{E}[\Psi_s].\end{aligned}$$

Noticing that $\bar{x} = \bar{x}_{s-1}$ and combining (50), we have

$$\begin{aligned}\mathbb{E}[\Phi_s] \leq{}& (1-\gamma\mu)^q \mathbb{E}\|x_{s-1}^q - x^*\|^2 - 2\gamma(1-4L\gamma)V_q \sum_{k=0}^{q-1} p_k \mathbb{E}\left[f(x_s^k) - f(x^*)\right] \\ &+ 12L\gamma^2 V_q \mathbb{E}\left[f(\bar{x}_{s-1}) - f(x^*)\right] + 24L\gamma^2 V_q \sum_{k=0}^{q-1} p_k \mathbb{E}\left[f(x_s^k) - f(x^*)\right] \\ ={}& (1-\gamma\mu)^q \mathbb{E}\|x_{s-1}^q - x^*\|^2 + \frac{1}{2}\mathbb{E}[\Psi_{s-1}] \\ &- 2\gamma(1-16L\gamma)V_q \sum_{k=0}^{q-1} p_k \mathbb{E}\left[f(x_s^k) - f(x^*)\right] \\ \leq{}& (1-\gamma\mu)^q \mathbb{E}\|x_{s-1}^q - x^*\|^2 + \frac{1}{2}\mathbb{E}[\Psi_{s-1}] \leq \rho\mathbb{E}[\Phi_{s-1}]\end{aligned}$$

where the second inequality holds due to $\gamma \leq 1/16L$ and $\rho = \max\{(1-\gamma\mu)^q, 1/2\}$. Recursively applying this inequality for s times outer loops, we have

$$\mathbb{E}[\Phi_s] \leq \rho^s \mathbb{E}[\Phi_0].$$

Since $\Psi_s \geq 0$, we can obtain

$$\mathbb{E}\|x_s^q - x^*\|^2 \leq \rho^s \mathbb{E}[\Phi_0].$$

Due to the L-smoothness of f, we have

$$\mathbb{E}\|x_s^q - x^*\|^2 \leq \rho^s (1 + 12L^2\gamma^2 V_q)\mathbb{E}\|x_0^q - x^*\|^2.$$

■

B.2 Proof of Corollary 22

Proof To obtain an ϵ-optimal solution, we should ensure

$$\rho^T(1+12L^2\gamma^2V_q)\mathbb{E}\|x_0^q-x^*\|^2\leq\epsilon.$$

Since $q=n$ and $\gamma=1/16L$, we can obtain

$$\rho^T\left(\frac{64+3V_q}{64}\right)\mathbb{E}\|x_0^q-x^*\|^2\leq\epsilon.$$

Rearranging the terms and taking logarithm on the both side, we have

$$T\ln\rho\leq\ln\frac{(64+3V_q)\mathbb{E}\|x_0^q-x^*\|^2}{64\epsilon},$$

which is equivalent to

$$T\geq\frac{1}{\ln(1/\rho)}\cdot\ln\frac{(64+3V_q)\mathbb{E}\|x_0^q-x^*\|^2}{64\epsilon}.$$

Recalling that $\rho=\max\{(1-\gamma\mu)^q,1/2\}$, then we have

$$\frac{1}{\ln(1/\rho)}=\max\left\{-\frac{1}{n}\cdot\frac{1}{\ln\left(1-\frac{1}{16\kappa}\right)},\frac{1}{\ln 2}\right\}. \tag{52}$$

Since $\ln x\leq x-1$ for all $x>0$, (52) can be upper bounded by

$$\max\left\{\frac{16\kappa}{n},2\right\}\geq\max\left\{-\frac{1}{n}\cdot\frac{1}{\ln\left(1-\frac{1}{16\kappa}\right)},\frac{1}{\ln 2}\right\}=\frac{1}{\ln(1/\rho)}.$$

Then as long as the iteration number for the outer loop T satisfies

$$T\geq\max\left\{\frac{16\kappa}{n},2\right\}\cdot\ln\frac{(64+3V_q)\mathbb{E}\|x_0^q-x^*\|^2}{64\epsilon},$$

GGD-WR-SVRG can achieve the ϵ-optimality. Noticing that within one outer loop, the number of partial derivative evaluations is $n(1+mb)d$. Then the total complexity is

$$2(1+bm)d\cdot\max\{8\kappa,n\}\ln\left(\frac{(64+3V_q)\mathbb{E}\|x_0^q-x^*\|^2}{64\epsilon}\right).$$

■

B.3 Proof of Theorem 23

Proof From the first inequality of (48) and the convexity of f, we know that

$$\begin{aligned}\mathbb{E}\|x_s^{k+1}-x^*\|^2 &\le \|x_s^k-x^*\|^2-2\gamma\left(1-4L\gamma\right)\Big(f(x_s^k)-f(x^*)\Big)+12L\gamma^2\left(f(\bar{x})-f(x^*)\right)\\ &\le \|x_s^k-x^*\|^2-2\gamma(1-10L\gamma)\Big(f(x_s^k)-f(x^*)\Big)+12L\gamma^2\left(f(\bar{x})-f(x^*)\right)\\ &\quad-12L\gamma^2\Big(f(x_s^k)-f(x^*)\Big).\end{aligned}$$

Rearranging the terms, taking the total expectation, we obtain

$$\begin{aligned}2\gamma(1-10L\gamma)\mathbb{E}\Big[f(x_s^k)-f(x^*)\Big] &\le \mathbb{E}\|x_s^k-x^*\|^2+12L\gamma^2\mathbb{E}\left[f(\bar{x})-f(x^*)\right]\\ &\quad-\mathbb{E}\|x_s^{k+1}-x^*\|^2-12L\gamma^2\mathbb{E}\Big[f(x_s^k)-f(x^*)\Big].\end{aligned}\tag{53}$$

Consider the option (b) in the outer loop and define Lyapunov function P^s as follows,

$$P^s \triangleq \mathbb{E}\left[\|x_{s+1}^0-x^*\|^2\right]+12L\gamma^2q\mathbb{E}\left[f(\bar{x}_s)-f(x^*)\right]\ge 0.$$

Sum (53) recursively over $k=0,1,...q-1$. We obtain

$$2\gamma(1-10L\gamma)\sum_{k=0}^{q-1}\mathbb{E}\Big[f(x_s^k)-f(x^*)\Big]\le P^{s-1}-P^s.$$

Summing over T outer loop, we have

$$2\gamma(1-10L\gamma)\sum_{s=1}^{T}\sum_{k=0}^{q-1}\mathbb{E}\Big[f(x_s^k)-f(x^*)\Big]\le P^0-P^T\le P^0.$$

Dividing qT on the both side, we obtain

$$\mathbb{E}\left[\frac{1}{qT}\sum_{s=1}^{T}\sum_{k=0}^{q-1}\Big(f(x_s^k)-f(x^*)\Big)\right]\le\frac{P^0}{2qT\gamma(1-10L\gamma)}.$$

Denoting $\hat{x}=\frac{1}{qT}\sum_{s=1}^{T}\sum_{k=0}^{q-1}x_s^k$, due to the convexity of the function f, we can conclude

$$\mathbb{E}\left[f(\hat{x})-f(x^*)\right]\le\frac{P^0}{2qT\gamma(1-10L\gamma)}=\frac{\mathbb{E}\|\bar{x}_0-x^*\|^2+12L\gamma^2q\mathbb{E}\left[f(\bar{x}_0)-f(x^*)\right]}{2qT\gamma(1-10L\gamma)}.\tag{54}$$

■

B.4 Proof of Corollary 24

Proof Substitute $\gamma=0.05/L$ into (54), we obtain

$$\mathbb{E}\left[f(\hat{x})-f(x^*)\right]\le\frac{20L\mathbb{E}\|\bar{x}_0-x^*\|^2+0.6q\mathbb{E}\left[f(x_0)-f(x^*)\right]}{qT}.$$

If $20L\mathbb{E}\|\bar{x}_0 - x^*\|^2/qT \leq \epsilon/2$ and $0.6\left[f(x_0) - f(x^*)\right]/T \leq \epsilon/2$, we can ensure that the expected optimality gap $\mathbb{E}\left[f(\hat{x}) - f(x^*)\right]$ is not greater than any given positive real number. To achieve ϵ-optimality, iteration number of outer loop T should satisfy

$$T \geq \max\left\{\frac{40L\mathbb{E}\|\bar{x}_0 - x^*\|^2}{n\epsilon}, \frac{1.2\mathbb{E}\left[f(\bar{x}_0) - f(x^*)\right]}{\epsilon}\right\}.$$

Since $q = n$, the total complexity is

$$nd(1+bm) \cdot \max\left\{\frac{40L\mathbb{E}\|\bar{x}_0 - x^*\|^2}{n\epsilon}, \frac{1.2\mathbb{E}\left[f(\bar{x}_0) - f(x^*)\right]}{\epsilon}\right\}.$$

■

B.5 Proof of Theorem 25

Proof Define Lyapunov function

$$R_s^k = \mathbb{E}\left[f(x_s^k) + \eta_k\|x_s^k - \bar{x}\|^2\right],$$

where η_k is defined in (11). From L-smoothness, we can obtain

$$f(x_s^{k+1}) \leq f(x_s^k) + \gamma\langle\nabla f(x_s^k), x_s^{k+1} - x_s^k\rangle + \frac{L}{2}\|x_s^{k+1} - x_s^k\|^2.$$

Taking expectation condition on x_s^k and all past, we have

$$\mathbb{E}\left[f(x_s^{k+1})\right] \leq \left[f(x_s^k)\right] - \gamma\|\nabla f(x_s^k)\|^2 + \frac{\gamma^2 L}{2}\mathbb{E}\|\tilde{g}_{m,b}^k\|^2. \tag{55}$$

Recalling the proof of Theorem 21, we can bound $\mathbb{E}\|\tilde{g}_{m,b}^k\|^2$ using (40) where the term $\mathbb{E}\|g_{m,b}(x_s^k) - g_{m,b}(\bar{x})\|^2$ can be bounded by

$$\begin{aligned}
\mathbb{E}\|g_{m,b}(x_s^k) - g_{m,b}(\bar{x})\|^2 &\leq \frac{n(m-1)}{bm(n-1)}\|\nabla f(x_s^k) - \nabla f(\bar{x})\|^2 \\
&\quad + \left(\frac{n-m}{bm(n-1)} + \frac{b-1}{b}\right)\frac{1}{n}\sum_{i=1}^n\|\nabla f_i(x_s^k) - \nabla f_i(\bar{x})\|^2 \\
&\leq \frac{n(m-1)}{bm(n-1)}L^2\|x_s^k - \bar{x}\|^2 \\
&\quad + \left(\frac{n-m}{bm(n-1)} + \frac{b-1}{b}\right)\frac{1}{n}\sum_{i=1}^n L^2\|x_s^k - \bar{x}\|^2 \\
&= L^2\|x_s^k - \bar{x}\|^2.
\end{aligned} \tag{56}$$

The term $\|\nabla f(\bar{x})\|^2$ can be bounded by

$$\|\nabla f(\bar{x})\|^2 \leq 2\|\nabla f(x_s^k) - f(\bar{x})\|^2 + 2\|\nabla f(x_s^k)\|^2 \leq 2L^2\|x_s^k - \bar{x}\|^2 + 2\|\nabla f(x_s^k)\|^2. \tag{57}$$

Combining (56) and (57), we can bound $\mathbb{E}\|\tilde{g}_{m,b}^k\|^2$ by

$$\mathbb{E}\|\tilde{g}_{m,b}^k\|^2 \leq 6L^2\|x_s^k - \bar{x}\|^2 + 4\|\nabla f(x_s^k)\|^2. \tag{58}$$

Now let us set this result aside, $\mathbb{E}\|x_s^{k+1} - \bar{x}\|^2$ can be bounded by

$$\begin{aligned}
\mathbb{E}\|x_s^{k+1} - \bar{x}\|^2 &= \mathbb{E}\|x_s^{k+1} - x_s^k + x_s^k - \bar{x}\|^2 \\
&= \gamma^2\mathbb{E}\|\tilde{g}_{m,b}^k\|^2 + \|x_s^k - \bar{x}\|^2 + 2\gamma\langle\nabla f(x_s^k), x_s^k - \bar{x}\rangle \\
&\leq \gamma^2\mathbb{E}\|\tilde{g}_{m,b}^k\|^2 + \|x_s^k - \bar{x}\|^2 + \frac{\gamma}{\tau}\|\nabla f(x_s^k)\| + \tau\gamma\|x_s^k - \bar{x}\|^2,
\end{aligned} \tag{59}$$

where the last inequality holds because

$$\forall\tau > 0,\, x,\, y \in \mathbb{R},\, xy \leq \frac{1}{2\tau}x^2 + \frac{\tau}{2}y^2. \tag{60}$$

From (55) and (59), taking total expectation, we can bound R_s^{k+1} by

$$\begin{aligned}
R_s^{k+1} &= \mathbb{E}\left[f(x_s^{k+1})\right] + \eta_{k+1}\mathbb{E}\|x_s^{k+1} - \bar{x}\|^2 \\
&\leq \mathbb{E}\left[f(x_s^k)\right] - \gamma\mathbb{E}\|\nabla f(x_s^k)\|^2 + \frac{\gamma^2 L}{2}\mathbb{E}\|\tilde{g}_{m,b}^k\|^2 \\
&+ \eta_{k+1}\left(\gamma^2\mathbb{E}\|\tilde{g}_{m,b}^k\|^2 + \mathbb{E}\|x_s^k - \bar{x}\|^2 + \frac{\gamma}{\tau}\mathbb{E}\|\nabla f(x_s^k)\|^2 + \tau\gamma\mathbb{E}\|x_s^k - \bar{x}\|^2\right).
\end{aligned}$$

Combining (58), we can obtain

$$\begin{aligned}
R_s^{k+1} &\leq \mathbb{E}\left[f(x_s^k)\right] - \gamma\mathbb{E}\|\nabla f(x_s^k)\|^2 + 3\gamma^2 L^3\mathbb{E}\|x_s^k - \bar{x}\|^2 + 2\gamma^2 L\mathbb{E}\|\nabla f(x_s^k)\|^2 \\
&+ \eta_{k+1}\gamma^2\left(6L^2\mathbb{E}\|x_s^k - \bar{x}\|^2 + 4\mathbb{E}\|\nabla f(x_s^k)\|^2\right) \\
&+ (1 + \tau\gamma)\eta_{k+1}\mathbb{E}\|x_s^k - \bar{x}\|^2 + \frac{\eta_{k+1}\gamma}{\tau}\mathbb{E}\|\nabla f(x_s^k)\|^2 \\
&= \mathbb{E}\left[f(x_s^k)\right] - \left(\gamma - 2\gamma^2 L - 4\gamma^2\eta_{k+1} - \frac{\eta_{k+1}\gamma}{\tau}\right)\mathbb{E}\|\nabla f(x_s^k)\|^2 \\
&+ \left(3\gamma^2 L^3 + 6\gamma^2 L^2\eta_{k+1} + (1 + \tau\gamma)\eta_{k+1}\right)\mathbb{E}\|x_s^k - \bar{x}\|^2 \\
&= R_s^k - \left(\gamma - 2\gamma^2 L - 4\gamma^2\eta_{k+1} - \frac{\eta_{k+1}\gamma}{\tau}\right)\mathbb{E}\|\nabla f(x_s^k)\|^2.
\end{aligned}$$

Rearranging the terms, we have

$$\left(\gamma - 2\gamma^2 L - 4\gamma^2\eta_{k+1} - \frac{\eta_{k+1}\gamma}{\tau}\right)\mathbb{E}\|\nabla f(x_s^k)\|^2 \leq R_s^k - R_s^{k+1}.$$

Since $\{\eta_k\}$ is decreasing, we can obtain

$$\left(\gamma - 2\gamma^2 L - 4\gamma^2\eta_0 - \frac{\eta_0\gamma}{\tau}\right)\mathbb{E}\|\nabla f(x_s^k)\|^2 \leq R_s^k - R_s^{k+1}.$$

Summing this recursion over q inner loops, noting that $\bar{x}_{s-1} = x_{s-1}^q = x_s^0$ and $\eta_q = 0$, we have

$$\left(\gamma - 2\gamma^2 L - 4\gamma^2\eta_0 - \frac{\eta_0\gamma}{\tau}\right)\sum_{k=0}^{q-1}\mathbb{E}\|\nabla f(x_s^k)\|^2 \leq R_s^0 - R_s^{k+1} = \mathbb{E}\left[f(\bar{x}_{s-1}) - f(\bar{x}_s)\right].$$

Summing this recursion over T outer loops, we have

$$\left(\gamma - 2\gamma^2 L - 4\gamma^2\eta_0 - \frac{\eta_0\gamma}{\tau}\right)\sum_{s=1}^{T}\sum_{k=0}^{q-1}\mathbb{E}\|\nabla f(x_s^k)\|^2 \leq \mathbb{E}\left[f(\bar{x}_0) - f(x_T^q)\right] \leq \mathbb{E}\left[f(\bar{x}_0) - f_{min}\right].$$

Dividing qT on the both side, rearranging the terms, we can obtain

$$\min_{\substack{k=0,1,\ldots,q-1\\ s=1,2,\ldots,T}} \mathbb{E}\|\nabla f(x_s^k)\|^2 \leq \frac{1}{qT}\sum_{s=1}^{T}\sum_{k=0}^{q-1}\mathbb{E}\|\nabla f(x_s^k)\|^2 \leq \frac{\mathbb{E}\left[f(x_1^0) - f_{min}\right]}{qT\gamma(1 - 2\gamma L - 4\gamma\eta_0 - \eta_0/\tau)}.$$

■

B.6 Proof of Corollary 26

Proof To achieve the ϵ-optimality, we need to ensure

$$\frac{\mathbb{E}\left[f(x_1^0) - f_{min}\right]}{qT\gamma(1 - 2\gamma L - 4\gamma\eta_0 - \eta_0/\tau)} \leq \epsilon.$$

Rearranging the terms, we have

$$T \geq \frac{\mathbb{E}\left[f(x_1^0) - f_{min}\right]}{\epsilon} \cdot \frac{1}{q\gamma} \cdot \frac{1}{1 - 2\gamma L - 4\gamma\eta_0 - \eta_0/\tau}. \tag{61}$$

The second term on the right side of (61) can be upper bounded by

$$\frac{1}{q\gamma} \leq \frac{7L}{n^{1/3}}. \tag{62}$$

η_0 can be upper bounded by

$$\begin{aligned}
\eta_0 = 3\gamma^2 L^3\frac{(1 + \tau\gamma + 6\gamma^2L^2)^q - 1}{\tau\gamma + 6\gamma^2L^2} &= \frac{3\psi^2 L}{n^{4/3}} \cdot \frac{\left(1 + \psi/n + 6\psi^2/n^{4/3}\right)^{\lceil n/7\psi\rceil} - 1}{\psi/n + 6\psi^2/n^{4/3}} \\
&= 3\psi^3 L\frac{\left(1 + \psi/n + 6\psi^2/n^{4/3}\right)^{\lceil n/7\psi\rceil} - 1}{\psi n^{1/3} + 6\psi^2} \\
&\leq 3\psi^2 L\frac{(1 + 7\psi/n)^{\lceil n/7\psi\rceil} - 1}{\psi n^{1/3}} \\
&\leq \frac{3\psi L(e-1)}{n^{1/3}},
\end{aligned}$$

where the first inequality holds because $\psi/n + 6\psi^2/n^{4/3} \leq 7\psi/n$ for $\psi < 1$ and $\psi n^{1/3} + 6\psi^2 > \psi n^{1/3}$, the last inequality holds because $(1 + 1/x)^x \leq e$. Combining the definition of ψ, we know that

$$\begin{aligned}
1 - 2\gamma L - 4\gamma\eta_0 - \eta_0/\tau &\geq 1 - \frac{2\psi}{n^{2/3}} - \frac{12\psi(e-1)}{n} - 3\psi(e-1) \\
&\geq \frac{3}{4} - 3\psi(e-1) \geq \frac{1}{2}.
\end{aligned} \tag{63}$$

Combining (62) and (63), we can upper bound the right side of (61) by

$$\frac{\mathbb{E}\left[f(x_1^0)-f_{min}\right]}{\epsilon}\cdot\frac{14L}{n^{1/3}}\geq\frac{\mathbb{E}\left[f(x_1^0)-f_{min}\right]}{\epsilon}\cdot\frac{1}{q\gamma}\cdot\frac{1}{1-2\gamma L-4\gamma\eta_0-\eta_0/\tau}.$$

Thus as long as we keep the iteration number of outer loop

$$T\geq\frac{\mathbb{E}\left[f(x_1^0)-f_{min}\right]}{\epsilon}\cdot\frac{14L}{n^{1/3}},$$

non-convex GGD-WR-SVRG can achieve ϵ-optimality, which indicates that the total complexity is

$$(n+\lceil n/7\psi\rceil\cdot mb)d\cdot\frac{14L\mathbb{E}\left[f(x_1^0)-f_{min}\right]}{\epsilon n^{1/3}}.$$

■

B.7 Proof of Theorem 28

Proof Most of this proof follows the analysis provided by Défossez et al. (2020) except that we derive the theoretical bound of GGD-WR-Adam using the original form of stepsize given by Kingma and Ba (2014) instead of a simplified stepsize given by Défossez et al. (2020). Denote

$$G_n=\nabla f(x^{n-1})\text{ and }\zeta_{n,(i)}=\frac{h_{n,(i)}}{\sqrt{\sigma+v_{n,(i)}}}\text{ and }\xi_{n,(i)}=\frac{g_{n,(i)}}{\sqrt{\sigma+v_{n,(i)}}},$$

and define $\tilde{v}_{n,k}\in\mathbb{R}^d$ as

$$\tilde{v}_{n,k,(i)}=\beta_2^k v_{n-k,(i)}+\mathbb{E}_{n-k-1}\left[\sum_{j=n-k+1}^{n}\beta_2^{n-j}g_{j,(i)}^2\right],$$

where $\mathbb{E}_{n-k-1}[\cdot]$ represents the expectation condition on all information before $n-k$-th iteration. Before deriving the theoretical bound for GGD-WR-Adam, we first prove some useful lemmas.

Lemma 33 *Suppose that Assumption 27 holds, the objective function f and the individual loss function f_i are L-smooth and $0<\beta_1<\beta_2<1$, then for all iterations $n\in\mathcal{N}^+$ generated by Algorithm 3, it satisfies*

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^{d}G_{n,(i)}\frac{h_{n,(i)}}{\sqrt{\sigma+v_{n,(i)}}}\right]\geq&\frac{1}{2}\left(\sum_{i=1}^{d}\sum_{k=0}^{n-1}\beta_1^k\mathbb{E}\left[\frac{G_{n-k,(i)}^2}{\sqrt{\sigma+\tilde{v}_{n,k+1,(i)}}}\right]\right)\\&-\gamma_{max}^2L^2\frac{(1-\beta_1)^{1/2}}{4R}\sum_{k=0}^{n-1}\beta_1^k\frac{k}{k+1}\sum_{l=1}^{k}\mathbb{E}\|\zeta_{n-l}\|^2\\&-\frac{3R}{(1-\beta)^{1/2}}\left(\sum_{k=0}^{n-1}\left(\frac{\beta_1}{\beta_2}\right)^k(k+1)\mathbb{E}\|\xi_{n-k}\|^2\right).\end{aligned}\tag{64}$$

Proof For some $n \in \mathcal{N}^+$, we have

$$\sum_{i=1}^{d} G_{n,(i)} \frac{h_{n,(i)}}{\sqrt{\sigma + v_{n,(i)}}} = \underbrace{\sum_{i=1}^{d} \sum_{k=0}^{n-1} \beta_1^k G_{n-k,(i)} \frac{g_{n-k,(i)}}{\sqrt{\sigma + v_{n,(i)}}}}_{\text{A}} + \underbrace{\sum_{i=1}^{d} \sum_{k=0}^{n-1} \beta_1^k \left(G_{n,(i)} - G_{n-k,(i)}\right) \frac{g_{n-k,(i)}}{\sqrt{\sigma + v_{n,(i)}}}}_{\text{B}}.$$

Let

$$\tau = \frac{(1-\beta_1)^{1/2}}{2R(k+1)},\ x = \frac{|g_{n-k,(i)}|}{\sqrt{\sigma + v_{n,(i)}}} \text{ and } y = |G_{n,(i)} - G_{n-k,(i)}|.$$

Using (60), we have

$$|B| \le \sum_{i=1}^{d} \sum_{k=0}^{n-1} \beta_1^k \left(\frac{(1-\beta_1)^{1/2}}{4R(k+1)} \left(G_{n,(i)} - G_{n-k,(i)}\right)^2 + \frac{R(k+1)}{(1-\beta_1)^{1/2}} \frac{g_{n-k,(i)}^2}{\sigma + v_{n,(i)}} \right).$$

Since $\sigma + v_{n,(i)} \ge \sigma + \beta_2^k v_{n-k,(i)} \ge \beta_2^k(\sigma + v_{n-k,(i)}$, we can obtain

$$|B| \le \sum_{i=1}^{d} \sum_{k=0}^{n-1} \beta_1^k \left(\frac{(1-\beta_1)^{1/2}}{4R(k+1)} \left(G_{n,(i)} - G_{n-k,(i)}\right)^2 + \frac{R(k+1)}{(1-\beta_1)^{1/2}} \frac{\xi_{n-k,(i)}^2}{\beta_2^k} \right). \tag{65}$$

From the L-smoothness of objective function f, we have

$$\|G_n - G_{n-k}\|^2 \le L^2 \|x^{n-1} - x^{n-k-1}\|^2 = L^2 \left\| \sum_{l=1}^{k} \gamma_{n-l} \zeta_{n-l} \right\|^2 \le \gamma_{max}^2 L^2 k \sum_{l=1}^{k} \|\zeta_{n-l}\|^2, \tag{66}$$

where $\gamma_{max} = \max\{\gamma_k\}$. Substituting (66) into (65), we have

$$\begin{aligned} |B| \le\ & \gamma_{max}^2 L^2 \sum_{k=0}^{n-1} \beta_1^k \frac{(1-\beta_1)^{1/2} k}{4R(k+1)} \sum_{l=1}^{k} \|\zeta_{n-l}\|^2 \\ & + \frac{R}{(1-\beta_1)^{1/2}} \sum_{k=0}^{n-1} \left(\frac{\beta_1}{\beta_2}\right)^k (k+1) \|\xi_{n-k}\|^2. \end{aligned} \tag{67}$$

Now we turn to deal with the term A. For simplicity, we drop the indices for now and denote $G = G_{n-k,(i)}$, $g = g_{n-k,(i)}$, $\tilde{v} = \tilde{v}_{n,k+1,(i)}$, $v = v_{n,(i)}$,

$$\theta^2 = \sum_{j=n-k}^{n} \beta_2^{n-j} g_{j,(i)}^2 \text{ and } r^2 = \mathbb{E}_{n-k-1} \theta^2.$$

Since $\tilde{v} - v = r^2 - \theta^2$, taking total expectation, we can rewrite the inside terms of term A as

$$\begin{aligned}\mathbb{E}\left[G\frac{g}{\sqrt{\sigma+v}}\right] &= \mathbb{E}\left[G\frac{g}{\sqrt{\sigma+\tilde{v}}} + Gg\left(\frac{1}{\sqrt{\sigma+v}} - \frac{1}{\sqrt{\sigma+\tilde{v}}}\right)\right] \\ &= \mathbb{E}\left[\mathbb{E}_{n-k-1}\left[G\frac{g}{\sqrt{\sigma+\tilde{v}}}\right] + Gg\frac{\theta^2-\delta^2}{\sqrt{\sigma+v}\sqrt{\sigma+\tilde{v}}(\sqrt{\sigma+v}+\sqrt{\sigma+\tilde{v}})}\right] \\ &= \mathbb{E}\left[\frac{G^2}{\sqrt{\sigma+\tilde{v}}}\right] + \mathbb{E}[\underbrace{Gg\frac{\theta^2-\delta^2}{\sqrt{\sigma+v}\sqrt{\sigma+\tilde{v}}(\sqrt{\sigma+v}+\sqrt{\sigma+\tilde{v}})}}_{C}]. \end{aligned} \tag{68}$$

Now we take a look at C,

$$|C| \leq \underbrace{|Gg|\frac{r^2}{(\sigma+v)^{1/2}(\sigma+\tilde{v})}}_{\text{D}} + \underbrace{|Gg|\frac{\theta^2}{(\sigma+v)(\sigma+\tilde{v})^{1/2}}}_{\text{E}},$$

due to the fact that $(\sigma+v)^{1/2} + (\sigma+\tilde{v})^{1/2} \geq \max\{(\sigma+v), (\sigma+\tilde{v})\}$ and $|r^2 - \theta^2| \leq r^2 + \theta^2$. Now using (60) again, if we let

$$\tau = \frac{\sqrt{1-\beta_1}\sqrt{\sigma+\tilde{v}}}{2},\ x = \frac{|g|r^2}{\sqrt{\sigma+\tilde{v}}\sqrt{\sigma+v}} \text{ and } y = \frac{|G|}{\sqrt{\sigma+\tilde{v}}},$$

we can obtain

$$D \leq \frac{G^2}{4\sqrt{\sigma+\tilde{v}}} + \frac{1}{\sqrt{1-\beta_1}}\frac{g^2r^4}{\sqrt{\sigma+\tilde{v}}\sqrt{\sigma+v}}.$$

Given that $\sigma + \tilde{v} \geq r^2$, taking conditional expectation, we have

$$\mathbb{E}_{n-k-1}[D] \leq \frac{G^2}{4\sqrt{\sigma+\tilde{v}}} + \frac{1}{\sqrt{1-\beta_1}}\frac{r^2}{\sqrt{\sigma+\tilde{v}}}\mathbb{E}_{n-k-1}\left[\frac{g^2}{\sigma+v}\right]. \tag{69}$$

Term E can be bounded in the similar way. If we let

$$\tau = \frac{\sqrt{1-\beta_1}\sqrt{\sigma+\tilde{v}}}{2r^2},\ x = \frac{|\theta g|}{\sigma+v} \text{ and } y = \frac{|G\theta|}{\sqrt{\sigma+\tilde{v}}},$$

we can obtain

$$E \leq \frac{G^2}{4\sqrt{\sigma+\tilde{v}}}\frac{\theta^2}{r^2} + \frac{1}{\sqrt{1-\beta_1}}\frac{r^2}{\sqrt{\sigma+\tilde{v}}}\frac{g^2\theta^2}{(\sigma+v)^2}. \tag{70}$$

Considering that $\sigma + v \geq \theta^2$ and $\mathbb{E}[\theta^2/r^2] = 1$, taking conditional expectation, we have

$$\mathbb{E}_{n-k-1}[E] \leq \frac{G^2}{4\sqrt{\sigma+\tilde{v}}} + \frac{1}{\sqrt{1-\beta_1}}\frac{r^2}{\sqrt{\sigma+\tilde{v}}}\mathbb{E}_{n-k-1}\left[\frac{g^2}{(\sigma+v)}\right]. \tag{71}$$

Noticing that in (70), we possibly divide by zero. It is suffice to notice that if $r^2 = 0$, then $\theta^2 = 0$ a.s. so that E = 0 and (71) still holds. Summing (69) and (71), we have

$$\mathbb{E}_{n-k-1}\left[|C|\right] \leq \frac{G^2}{2\sqrt{\sigma+\tilde{v}}} + \frac{2}{\sqrt{1-\beta_1}}\frac{r^2}{\sigma+\tilde{v}}\mathbb{E}_{n-k-1}\left[\frac{g^2}{\sigma+v}\right].$$

Since $r \leq \sqrt{\sigma + \tilde{v}}$, and from the definition of r, we know that $r \leq (k+1)R$. Reintroducing the indices and noticing that $\sigma + v_{n,(i)} \geq \sigma + \beta_2^k v_{n-k,(i)} \geq \beta_2^k(\sigma + v_{n-k,(i)})$, we can obtain after taking total expectation,

$$\mathbb{E}[|C|] \leq \mathbb{E}\left[\frac{1}{2}\frac{G_{n-k,(i)}^2}{\sqrt{\sigma + \tilde{v}_{n,k+1,(i)}}}\right] + \frac{2R(k+1)}{(1-\beta_1)^{1/2}\beta_2^k}\mathbb{E}\left[\frac{g_{n-k,(i)}^2}{\sigma + v_{n-k,(i)}}\right]. \tag{72}$$

Substituting (72) into (68), we have

$$\begin{aligned}\mathbb{E}[A] &\geq \sum_{i=1}^{d}\sum_{k=0}^{n-1}\left(\mathbb{E}\left[\frac{G_{n-k,i}^2\beta_1^k}{\sqrt{\sigma + \tilde{v}_{n,k+1,i}}}\right] - \left(\frac{1}{2}\mathbb{E}\left[\frac{G_{n-k,i}^2\beta_1^k}{\sqrt{\sigma + \tilde{v}_{n,k,i}}}\right] + \frac{2R(k+1)}{\sqrt{1-\beta_1}\beta_2^k}\mathbb{E}\left[\frac{g_{n-k,i}^2\beta_1^k}{\sigma + v_{n-k,i}}\right]\right)\right) \\ &= \frac{1}{2}\left(\sum_{i=1}^{d}\sum_{k=0}^{n-1}\beta_1^k\mathbb{E}\left[\frac{G_{n-k,i}^2}{\sqrt{\sigma + \tilde{v}_{n,k+1,i}}}\right]\right) - \frac{2R}{\sqrt{1-\beta_1}}\left(\sum_{i=1}^{d}\sum_{k=0}^{n-1}\left(\frac{\beta_1}{\beta_2}\right)^k(k+1)\mathbb{E}\left\|\xi_{n-k}\right\|^2\right).\end{aligned} \tag{73}$$

Combining (67) and (73), we prove Lemma 33. ■

Lemma 34 *Suppose that $0 < \beta_1 < \beta_2 \leq 1$. For a sequence of real numbers $\{a_n\}$, denoting $b_n = \sum_{j=1}^{n}\beta_2^{n-j}a_j^2$ and $c_n = \sum_{j=1}^{n}\beta_1^{n-j}a_j$, then we have*

$$\sum_{j=1}^{n}\frac{c_j^2}{\sigma + b_j} \leq \frac{1}{(1-\beta_1)(1-\beta_1/\beta_2)}\left(\ln\left(1 + \frac{b_n}{\sigma}\right) - n\ln(\beta_2)\right)$$

Proof For some $j \in \mathcal{N}^+$, $j \leq n$, using Jensen inequality, we have

$$c_j^2 \leq \frac{1}{1-\beta_1}\sum_{l=1}^{j}\beta_1^{j-l}a_l^2,$$

so that

$$\frac{c_j^2}{\sigma + b_j} \leq \frac{1}{1-\beta_1}\sum_{l=1}^{j}\beta_1^{j-l}\frac{a_l^2}{\sigma + b_j}.$$

From the definition of b_l, for any $l \leq j$, we know that $\sigma + b_j \geq \sigma + \beta_2^{j-l}b_l \geq \beta_2^{j-l}(\sigma + b_l)$. Then we can obtain

$$\frac{c_j^2}{\sigma + b_j} \leq \frac{1}{1-\beta_1}\sum_{l=1}^{j}\left(\frac{\beta_1}{\beta_2}\right)^{j-l}\frac{a_l^2}{\sigma + b_l}.$$

Summing over all $j \in [n]$, we have

$$\begin{aligned}\sum_{j=1}^{n}\frac{c_j^2}{\sigma + b_j} &\leq \frac{1}{1-\beta_1}\sum_{j=1}^{n}\sum_{l=1}^{j}\left(\frac{\beta_1}{\beta_2}\right)^{j-l}\frac{a_l^2}{\sigma + b_l} \\ &= \frac{1}{1-\beta_1}\sum_{l=1}^{n}\frac{a_l^2}{\sigma + b_l}\sum_{j=l}^{n}\left(\frac{\beta_1}{\beta_2}\right)^{j-l} \\ &\leq \frac{1}{(1-\beta_1)(1-\beta_1/\beta_2)}\sum_{l=1}^{n}\frac{a_l^2}{\sigma + b_l}.\end{aligned} \tag{74}$$

The main term in summation (74) can be bounded by

$$\begin{aligned}\frac{a_l^2}{\sigma+b_l} &\le \ln(\sigma+b_l)-\ln(\sigma+b_l-a_l^2)\\ &= \ln(\sigma+b_l)-\ln(\sigma+\beta_2 b_{l-1})\\ &= \ln\left(\frac{\sigma+b_l}{\sigma+b_{l-1}}\right)+\ln\left(\frac{\sigma+b_{l-1}}{\sigma+\beta_2 b_{l-1}}\right)\\ &\le \ln\left(\frac{\sigma+b_l}{\sigma+b_{l-1}}\right)-\ln(\beta_2).\end{aligned}$$

Then (74) can be further bounded by

$$\begin{aligned}\sum_{j=1}^{n}\frac{c_j^2}{\sigma+b_j} &\le \frac{1}{(1-\beta_1)(1-\beta_1/\beta_2)}\sum_{l=1}^{n}\frac{a_l^2}{\sigma+b_l}\\ &\le \frac{1}{(1-\beta_1)(1-\beta_1/\beta_2)}\left(\ln\left(1+\frac{b_n}{\sigma}\right)-n\ln(\beta_2)\right),\end{aligned}$$

which proves Lemma 34. ■

Lemma 35 *Given $0<x<1$, we have*

$$\sum_{n=0}^{\infty}x^n n=\frac{x}{(1-x)^2}.$$

Proof Since $0<x<1$, we know that the infinite series $\sum_{n=0}^{\infty}x^n n$ is convergent. Denote $S(x)=\sum_{n=0}^{\infty}x^n n$, then for any $0<x<1$, dividing x on the both side, then we have

$$\int_0^s\frac{S(x)}{x}\mathrm{d}x=\sum_{n=0}^{\infty}\int_0^s nx^{n-1}\mathrm{d}x=\sum_{n=0}^{\infty}s^n=\frac{s}{1-s}.$$

Taking derivatives with respect to s, we have

$$\frac{S(s)}{s}=\frac{\partial\left(\frac{s}{1-s}\right)}{\partial s}=\frac{1}{(1-s)^2},$$

which proves Lemma 35. ■

Having finished the proof of three useful lemmas, we now formally prove Theorem 28. For some $n\in\mathcal{N}^+$, using L-smoothness of objective function f, we have

$$f(x^n)\le f(x^{n-1})-\gamma_n G_n^\top\zeta_n+\frac{\gamma_n^2 L}{2}\|\zeta_n\|^2.$$

Taking total expectation and using Lemma 33, we have

$$\begin{aligned}\mathbb{E}\left[f\left(x^n\right)\right] \leq \mathbb{E}\left[f\left(x^{n-1}\right)\right] - \frac{\gamma_n}{2}\left(\sum_{i=1}^{d}\sum_{k=0}^{n-1}\beta_1^k\mathbb{E}\left[\frac{G_{n-k,(i)}^2}{2\sqrt{\sigma+\tilde{v}_{n,k+1,(i)}}}\right]\right) + \frac{\gamma_n^2 L}{2}\mathbb{E}\left[\left\|\zeta_n\right\|^2\right] \\ + \frac{\gamma_{max}^3 L^2\sqrt{1-\beta_1}}{4R}\left(\sum_{l=1}^{n-1}\mathbb{E}\left\|\zeta_{n-l}\right\|^2\sum_{k=l}^{n-1}\beta_1^k\frac{k}{k+1}\right) \\ + \frac{3\gamma_n R}{\sqrt{1-\beta_1}}\left(\sum_{k=0}^{n-1}\left(\frac{\beta_1}{\beta_2}\right)^k(k+1)\mathbb{E}\left\|\xi_{n-k}\right\|^2\right).\end{aligned}$$

Since for any $k \in \mathcal{N}$ and $k < n$, we have $\sqrt{\sigma+\tilde{v}_{n,k+1,(i)}} \leq R\sqrt{\sum_{j=0}^{n-1}\beta_2^j}$. Denoting $\Omega_n = \sqrt{\sum_{j=0}^{n-1}\beta_2^j}$, we have

$$\begin{aligned}\mathbb{E}\left[f\left(x^n\right)\right] \leq \mathbb{E}\left[f\left(x^{n-1}\right)\right] - \frac{\gamma_n}{2R\Omega_n}\left(\sum_{k=0}^{n-1}\beta_1^k\mathbb{E}\|G_{n-k}\|^2\right) + \frac{\gamma_n^2 L}{2}\mathbb{E}\left[\left\|\zeta_n\right\|_2^2\right] \\ + \frac{\gamma_{max}^3 L^2\sqrt{1-\beta_1}}{4R}\left(\sum_{l=1}^{n-1}\mathbb{E}\left\|\zeta_{n-l}\right\|_2^2\sum_{k=l}^{n-1}\beta_1^k\frac{k}{k+1}\right) \\ + \frac{3\gamma_n R}{\sqrt{1-\beta_1}}\left(\sum_{k=0}^{n-1}\left(\frac{\beta_1}{\beta_2}\right)^k(k+1)\mathbb{E}\left\|\xi_{n-k}\right\|_2^2\right).\end{aligned}$$

Summing over all T iterations, rearranging the terms, noticing that the objective function is bounded below by f_{min}, we have

$$\begin{aligned}\underbrace{\frac{1}{2R}\sum_{n=1}^{T}\frac{\gamma_n}{\Omega_n}\sum_{k=0}^{n-1}\beta_1^k\mathbb{E}\left\|G_{n-k}\right\|^2}_{\text{F}} \leq \mathbb{E}\left(f\left(x^0\right) - f_{min}\right) + \underbrace{\frac{\gamma_{max}^2 L}{2}\sum_{n=1}^{T}\mathbb{E}\left\|\zeta_n\right\|^2}_{\text{G}} \\ + \underbrace{\frac{\gamma_{max}^3 L^2(1-\beta_1)^{1/2}}{4R}\sum_{n=1}^{T}\sum_{l=1}^{n-1}\mathbb{E}\left\|\zeta_{n-l}\right\|^2\sum_{k=l}^{n-1}\beta_1^k\frac{k}{k+1}}_{\text{H}} \\ + \underbrace{\frac{3\gamma_{max}R}{(1-\beta_1)^{1/2}}\sum_{n=1}^{T}\sum_{k=0}^{n-1}\left(\frac{\beta_1}{\beta_2}\right)^k(k+1)\mathbb{E}\left\|\xi_{n-k}\right\|^2}_{\text{I}}. \qquad (75)\end{aligned}$$

Now we proceed these terms sequentially. Note that γ_n/Ω_n in term F can be lower bounded by

$$\frac{\gamma_n}{\Omega_n} = \gamma\frac{1-\beta_1}{(1-\beta_2)^{1/2}}\frac{(1-\beta_2^n)^{1/2}}{1-\beta_n}\frac{(1-\beta_2)^{1/2}}{(1-\beta_2^n)^{1/2}} = \gamma\frac{1-\beta_1}{1-\beta_1^n} \geq \gamma(1-\beta_1).$$

Using the change of index $j = n - k$, we can obtain

$$\begin{aligned}
F &= \frac{1}{2R}\sum_{n=1}^{T}\frac{\gamma_n}{\Omega_n}\sum_{j=1}^{n}\beta_1^{n-j}\mathbb{E}\left\|G_j\right\|^2 \\
&\geq \frac{\gamma\left(1-\beta_1\right)}{2R}\sum_{j=1}^{T}\mathbb{E}\left\|G_j\right\|^2\sum_{n=j}^{T}\beta_1^{n-j} \\
&= \frac{\gamma}{2R}\sum_{j=1}^{T}\left(1-\beta_1^{T-j+1}\right)\mathbb{E}\left\|G_j\right\|^2 \\
&= \frac{\gamma}{2R}\sum_{j=1}^{T}\left(1-\beta_1^{T-j+1}\right)\mathbb{E}\left\|\nabla f\left(x^{j-1}\right)\right\|^2 \\
&= \frac{\gamma}{2R}\sum_{i=0}^{T-1}\left(1-\beta_1^{T-j}\right)\mathbb{E}\left\|\nabla f\left(x^{j}\right)\right\|^2.
\end{aligned}$$

Considering the random index ω defined in Theorem 28 and noticing that

$$\sum_{j=0}^{T-1}(1-\beta_1^{T-j}) = T - \beta_1\frac{1-\beta_1^T}{1-\beta_1} \geq T - \frac{\beta_1}{1-\beta_1} = \tilde{T},$$

we have

$$F \geq \frac{\gamma\tilde{T}}{2R}\mathbb{E}\|\nabla f(x^{\omega})\|^2. \tag{76}$$

Then we take a look at term G. Using Lemma 34, we have

$$G \leq \frac{\gamma_{max}^2 L}{2(1-\beta_1)(1-\beta_1/\beta_2)}\sum_{i=1}^{d}\left(\ln\left(1+\frac{v_{T,(i)}}{\sigma}\right) - T\ln(\beta_2)\right).$$

Noting that γ_{max} is equivalent to

$$\begin{aligned}
\gamma_n^2 = \gamma^2\frac{(1-\beta_1)^2}{(1-\beta_2)}\frac{1-\beta_2^n}{(1-\beta_1^n)^2} &\leq \gamma^2\frac{(1-\beta_1)^2}{(1-\beta_2)}\frac{1}{(1-\beta_1^n)^2} \\
&\leq \gamma^2\frac{1}{(1-\beta_2)(1+\beta_1)^2} \leq \frac{\gamma^2}{1-\beta_2} = \gamma_{max}^2,
\end{aligned}$$

where the last inequality holds because $(1-\beta_1^n)^2 = (1-\beta_1)^2(\sum_{k=0}^{n-1}\beta_1^k)^2 \geq (1-\beta_1)^2(1+\beta_1)^2$, we can thus obtain

$$G \leq \frac{\gamma^2 L}{2(1-\beta_1)(1-\beta_2)(1-\beta_1/\beta_2)}\sum_{i=1}^{d}\left(\ln\left(1+\frac{v_{T,(i)}}{\sigma}\right) - T\ln(\beta_2)\right). \tag{77}$$

Next we proceed with term H. Replacing the index j with $n-l$, we have

$$\begin{aligned}
H &= \frac{\gamma_{max}^3 L^2 (1-\beta_1)^{1/2}}{4R} \sum_{n=1}^{T}\sum_{j=1}^{n} \mathbb{E}\left\|\zeta_{n-l}\right\|^2 \sum_{k=n-j}^{n-1} \beta_1^k \frac{k}{k+1} \\
&= \frac{\gamma_{max}^3 L^2 (1-\beta_1)^{1/2}}{4R} \sum_{j=1}^{T} \mathbb{E}\|\zeta_j\|^2 \sum_{n=j}^{T}\sum_{k=n-j}^{n-1} \beta_1^k \frac{k}{k+1} \\
&= \frac{\gamma_{max}^3 L^2 (1-\beta_1)^{1/2}}{4R} \sum_{j=1}^{T} \mathbb{E}\|\zeta_j\|^2 \sum_{k=0}^{T-1} \beta_1^k \frac{k}{k+1} \sum_{n=j}^{j+k} 1 \\
&= \frac{\gamma_{max}^3 L^2 (1-\beta_1)^{1/2}}{4R} \sum_{j=1}^{T} \mathbb{E}\|\zeta_j\|^2 \sum_{k=0}^{T-1} \beta_1^k k \\
&\le \frac{\gamma_{max}^3 L^2 (1-\beta_1)^{1/2}}{4R} \frac{\beta_1}{(1-\beta_1)^2} \sum_{j=1}^{T} \mathbb{E}\|\zeta_j\|^2 \\
&\le \frac{\gamma^3 L^2}{4R} \frac{(1-\beta_1)^{1/2}}{(1-\beta_2)^{3/2}} \frac{1}{(1-\beta_1)^2} \sum_{j=1}^{T} \mathbb{E}\|\zeta_j\|^2 = \frac{\gamma^3 L^2}{4R} \frac{1}{(1-\beta_2)^{3/2}(1-\beta_1)^{3/2}} \sum_{j=1}^{T} \mathbb{E}\|\zeta_j\|^2,
\end{aligned}$$

where the first inequality holds due to Lemma 35. Using Lemma 34 again, we can obtain

$$\begin{aligned}
H &\le \frac{\gamma^3 L^2}{4R} \frac{1}{(1-\beta_2)^{3/2}(1-\beta_1)^{3/2}} \sum_{j=1}^{T} \mathbb{E}\|\zeta_j\|^2 \\
&\le \frac{\gamma^3 L^2}{4R} \frac{1}{(1-\beta_2)^{3/2}(1-\beta_1)^{5/2}(1-\beta_1/\beta_2)} \sum_{i=1}^{d} \left(\ln\left(1+\frac{v_{T,(i)}}{\sigma}\right) - T\ln(\beta_2)\right). \qquad (78)
\end{aligned}$$

Finally, we can move on to term I. Replacing index j with $n-k$, we can obtain

$$\begin{aligned}
I &\le \frac{3\gamma_{max} R}{(1-\beta_1)^{1/2}} \sum_{n=1}^{T}\sum_{k=0}^{n-1} \left(\frac{\beta_1}{\beta_2}\right)^k (k+1)\mathbb{E}\left\|\xi_{n-k}\right\|^2 \\
&\le \frac{3\gamma_{max} R}{(1-\beta_1)^{1/2}} \sum_{j=1}^{T} \mathbb{E}\|\xi_j\|^2 \sum_{n=j}^{T} \left(\frac{\beta_1}{\beta_2}\right)^{n-j} (n-j+1) \\
&\le \frac{3\gamma R}{(1-\beta_1)^{1/2}(1-\beta_2)^{1/2}(1-\beta_1/\beta_2)^2} \sum_{j=1}^{T} \mathbb{E}\|\xi_j\|^2.
\end{aligned}$$

Again using Lemma 34, we can obtain

$$I \le \frac{3\gamma R}{(1-\beta_1)^{3/2}(1-\beta_2)^{1/2}(1-\beta_1/\beta_2)^3} \sum_{i=1}^{d} \left(\ln\left(1+\frac{v_{T,(i)}}{\sigma}\right) - T\ln(\beta_2)\right) \qquad (79)$$

Noticing that we have $v_{T,(i)} \le R^2/1-\beta_2$ for any $i \in [d]$, substituting (76), (77), (78), (79) into (75), we can derive the desired result

$$\mathbb{E}\|\nabla f(x^\omega)\|^2 \le \frac{2R\mathbb{E}\left(f(x^0) - f_{min}\right)}{\gamma\tilde{T}} + \frac{J}{\tilde{T}}\left(\ln\left(1+\frac{R^2}{\sigma(1-\beta_2)}\right) - T\ln(\beta_2)\right),$$

where

$$J = \frac{\gamma dRL}{(1-\beta_1)(1-\beta_2)(1-\beta_1/\beta_2)} + \frac{\gamma^2 dL^2}{2(1-\beta_2)^{3/2}(1-\beta_1)^{5/2}(1-\beta_1/\beta_2)} + \frac{6dR^2}{(1-\beta_1)^{3/2}(1-\beta_2)^{1/2}(1-\beta_1/\beta_2)^3}.$$

■

B.8 Proof of Corollary 29

Proof From (12), we know that as long as

$$2R\frac{\mathbb{E}\left[f(x^0)-f_{min}\right]}{\tilde{\gamma}\sqrt{T}} \leq \frac{\epsilon}{3},\ \frac{K}{\sqrt{T}}\ln\left(1+\frac{TR^2}{\sigma}\right) \leq \frac{\epsilon}{3} \text{ and } \frac{K}{\sqrt{T}} \leq \frac{\epsilon}{3}, \tag{80}$$

we can ensure the ϵ-optimality. The first inequality in (80) indicates that

$$T \geq \frac{36R^2\left[\mathbb{E}f(x^0)-f_{min}\right]^2}{\tilde{\gamma}^2\epsilon^2}. \tag{81}$$

The left side of second inequality in (80) can be upper bound by

$$\begin{aligned}
\frac{K}{\sqrt{T}}\ln\left(1+\frac{TR^2}{\sigma}\right) &= K\frac{\ln\left(1+\frac{TR^2}{\sigma}\right)}{\sqrt{1+\frac{TR^2}{\sigma}}}\left(\frac{1}{T}+\frac{R^2}{\sigma}\right)^{1/2} \leq K\left(1+\frac{R^2}{\sigma}\right)^{1/2}\frac{\ln\left(1+\frac{TR^2}{\sigma}\right)}{\sqrt{1+\frac{TR^2}{\sigma}}} \\
&\leq K\left(1+\frac{R^2}{\sigma}\right)^{1/2}\frac{\ln\left(1+\frac{TR^2}{\sigma}\right)}{(1+\frac{TR^2}{\sigma})^{\phi}}\cdot\frac{1}{(1+\frac{TR^2}{\sigma})^{1/2-\phi}} \\
&\leq \frac{K}{\phi e}\left(1+\frac{R^2}{\sigma}\right)^{1/2}\cdot\frac{1}{(1+\frac{TR^2}{\sigma})^{1/2-\phi}},
\end{aligned}$$

where the last inequality holds due to $\ln(x)/x^{\phi} \leq 1/(\phi e)$ for $\phi \in (0,1/2)$. Thus we only need to ensure that

$$\frac{3K}{\phi e\epsilon}\left(1+\frac{R^2}{\sigma}\right)^{1/2} \leq (1+\frac{TR^2}{\sigma})^{1/2-\phi},$$

which is equivalent to

$$T \geq \frac{\sigma}{R^2}\left(\left(\frac{3K}{\phi e\epsilon}\right)^{\frac{2}{1-2\phi}}\cdot\left(1+\frac{R^2}{\sigma}\right)^{\frac{1}{1-2\phi}}-1\right). \tag{82}$$

Since the second inequality can be bounded by $\epsilon/3$, the third inequality can be bounded by $\epsilon/3$ as well. Combining (81) and (82), we have

$$T \geq \max\left\{\frac{36R^2\left[\mathbb{E}f(x^0)-f_{min}\right]^2}{\tilde{\gamma}^2\epsilon^2}, \frac{\sigma}{R^2}\left(\left(\frac{3K}{\phi e\epsilon}\right)^{\frac{2}{1-2\phi}}\cdot\left(1+\frac{R^2}{\sigma}\right)^{\frac{1}{1-2\phi}}-1\right)\right\}.$$

■

Data set	GGD	MBSGD	SGD-IS	SVRG	GSVRG	Adam	GAdam
covtype	1,288.7	1,077.8	4,512.5	2,249.6	3,042.1	1,002.6	1,388.5
ijcnn1	302.8	230.3	527.2	369.1	496.28	280.1	373.3
a9a	165.7	118.4	318.7	318.7	386.6	125.8	172.6
rcv1	649.7	388.3	762.9	1,320.3	2,280.6	516.5	796.8
MNIST	942.1	766.9	**	988.2	1,835.4	721.2	982.2
CIFAR-10	9,849.7	7,718.2	**	15,361.5	19,981.2	8,026.5	10,662.3

Table 4: Summary of the execution time (s)

Appendix C. Additional Experiment Results and Network Architectures

In Appendix C, we present some additional experiment results to complement our discussions and provide the convolution neural network architectures used in Section 6.

C.1 Execution Time and Time Plots

All the experiments are carried out on a personal computer (3.70 GHz 12th Gen Intel Core i5 with 16 GB RAM and NVIDIA RTX 3080). Table 4 presents the execution time of our programs. From these results, we can see that in general, the increasing in the size of data sets and the complexity of model, may lead to a longer time that CPU and GPU charge for execution. The execution time for training CNNs using SGD-IS is not recorded here as the importance sampling probability is not available.

To further explore the relation of execution time and the performance, we present the time plots which depict the trade offs between the execution time (x-axis) and the train loss, square norm of full gradient and test loss (y-axis) respectively in Figures 12 and 13 where we sketchily show the efficiency of different algorithms in practical use. From Figure 12 and Table 4, we can see that for GAdam and Adam, since grafting gradient based methods take more time to finish one epoch of training, GAdam is little worse than Adam at the beginning. However, after about 500 seconds of training, GAdam catches up with Adam and their performances become quite indistinguishable. The comparison between GSVRG and SVRG is slightly different in two counts. One is that GSVRG achieves a lower train loss after about 750 seconds of training and the other is that SVRG achieves a much lower test loss. When comparing GGD with MBSGD and SGD-IS, we find that although GGD decreases slower than those two methods, it achieves comparable performances after 700 seconds of training. For training cifar10-nv on *CIFAR-10* data set, generally speaking, grafting gradient based methods are more efficient than stochastic gradient based methods. Especially in Figures 13(g) and 13(i), GGD-WR-SVRG shows significant improvement over SVRG. Combining these two figures and the results in Section 6, we can conclude that in practice, it is worth using grafting gradient to update model parameters as its improvements over stochastic gradient based methods can compensate the time cost to some extent.

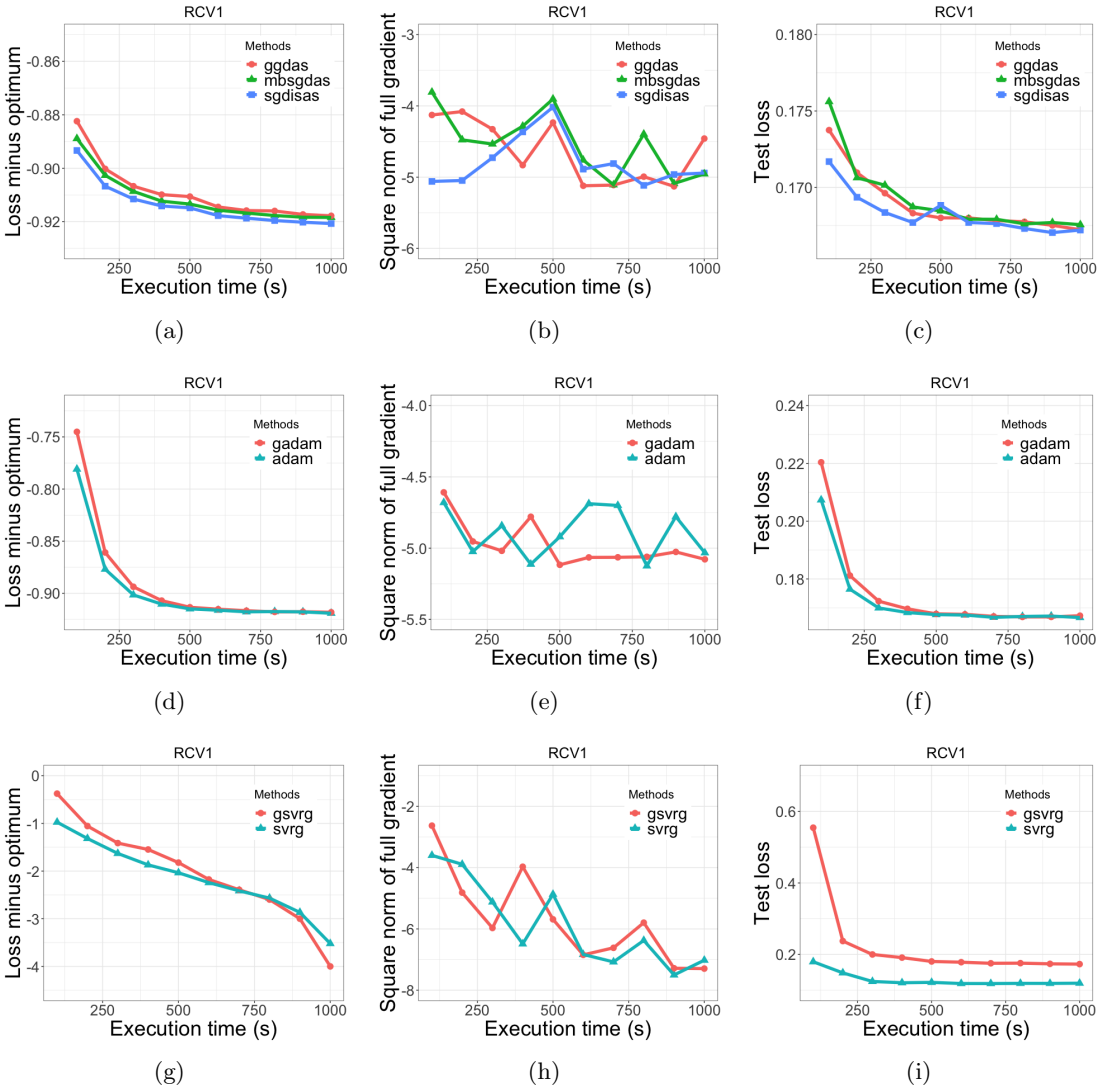


Figure 12: Comparisons of the train loss (left), square norm of full gradient (middle) and test loss (right) on rcv1.

C.2 Impact of Minibatching on GGD and SGD-IS

We also run the addtional experiments which are used to compare the impact of the subsampled set size m on GGD and SGD with importance sampling. For the latter one, we use the average of gradients which are sampled from population with probability $P_{r_i} \propto L_{r_i}$ to update the model parameters. These additional experiments are run to solve the same L_2-regularized logistic regression problems on *ijcnn* and *a9a* data sets. For fair comparison, t-inverse learning schedule is used both for mini-batch SGD with importance sampling and GGD, the learning rates of two methods with same subsampled set size m are set to be the same. We present the empirical results in Figures 14 and 15.

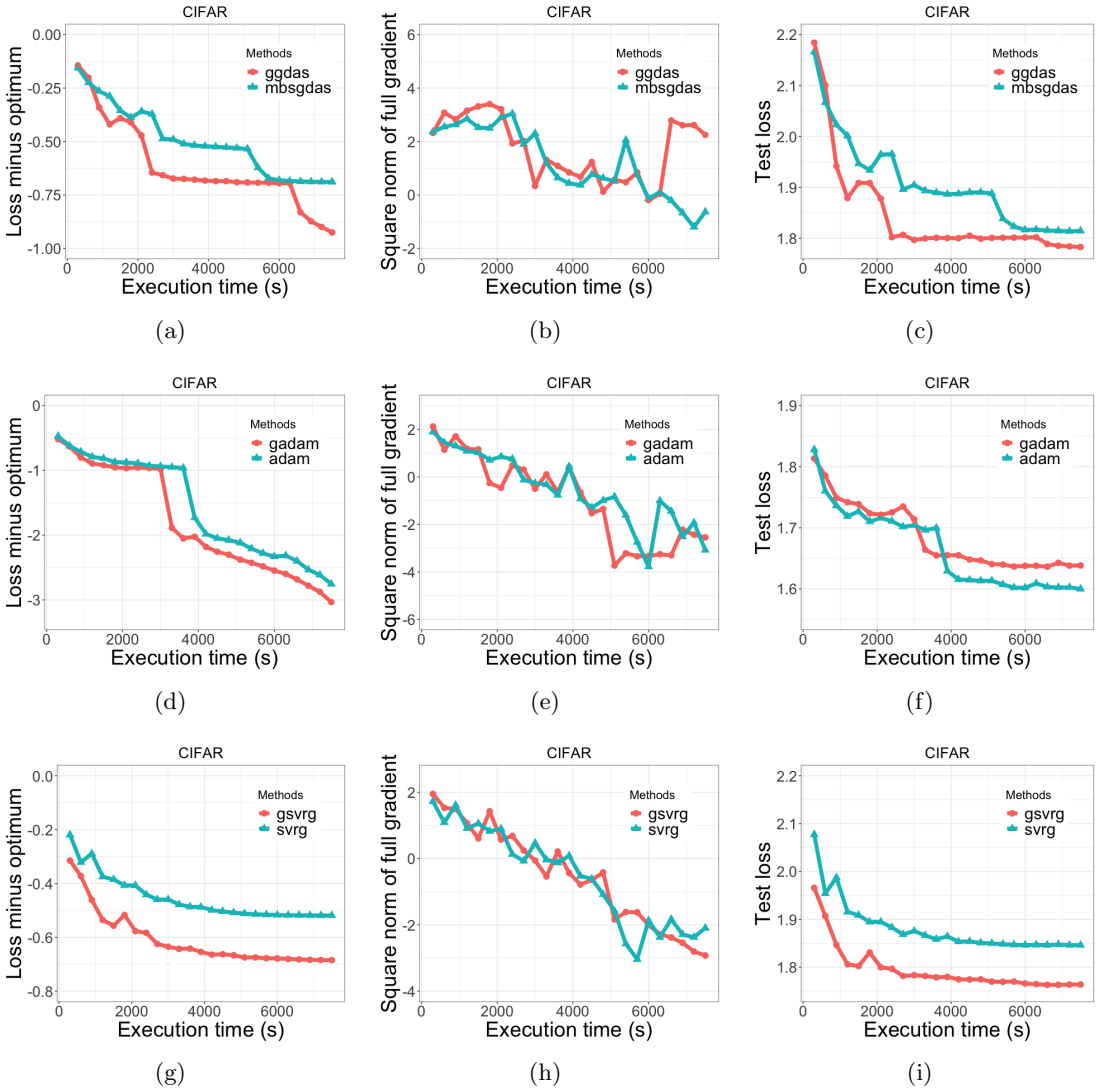


Figure 13: Comparisons of the train loss (left), square norm of full gradient (middle) and test loss (right) on CIFAR-10.

The suffix $-2\hat{}k$ for $k \in \{4, 5, 6, 7\}$ represents the subsampled set size m (ranging from 16 to 128) used in algorithms. From these results, we can see that when the performance of MBSGD is quite close to the performance of SGD with importance sampling as shown in Figure 4(b), GGD does not benefit a lot from increasing mini-batch size as shown in Figure 15. Coincidentally from Figure 14 we can see that GGD with larger subsampled set size m can achieve a lower train loss and test loss when the performance of SGD-IS is better than the performance of mini-batch SGD. As for SGD with importance sampling, results in Figures 14 and 15 show that SGD with importance seems to be insusceptible to minibatching technique. Combining Figures 14, 15 and the corresponding results in Figure

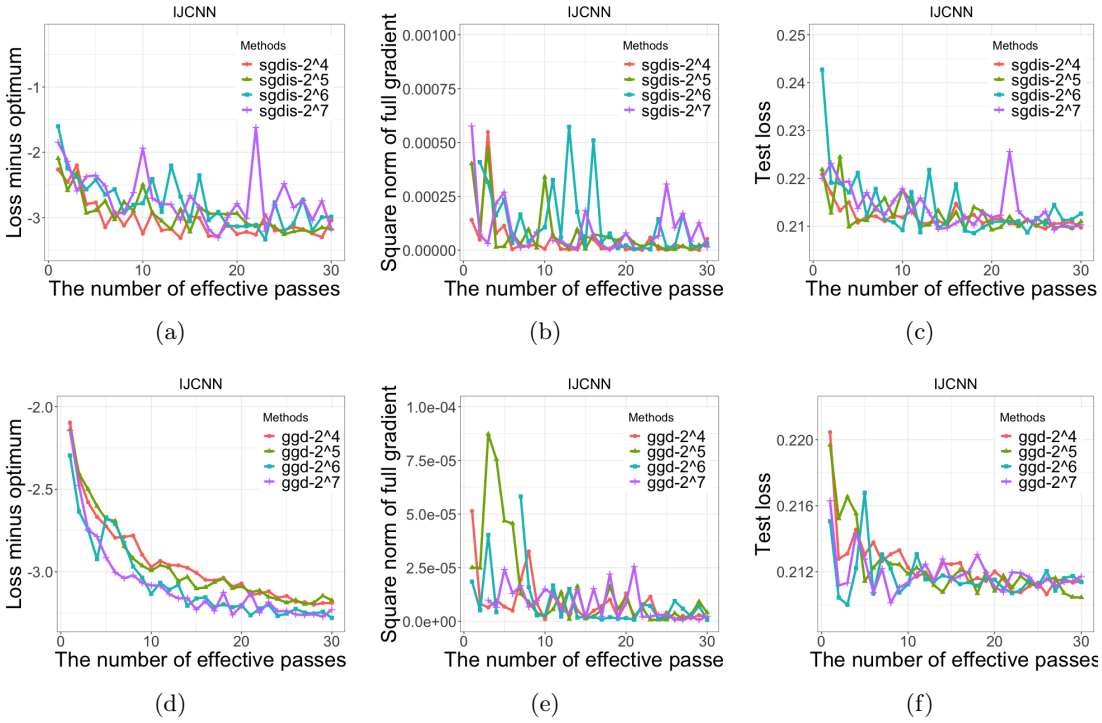


Figure 14: Comparisons of the train loss (left), square norm of full gradient (middle) and test loss (right) on ijcnn.

4, we can conclude that when mini-batch SGD performs quite similarily to SGD with importance sampling, that is, when using minibatching technique is not more profitable than using importance sampling technique solely, GGD does not benefit from minibatching technique more than SGD with importance sampling does. On the contrary, when SGD with importance sampling outperforms the mini-batch SGD, minibatching technique seems to be more impactful when applying to GGD as increasing subsampled set size m can results in the improvement of GGD methods. This is expected since increasing subsampled set size m can reduce the noise variance of mini-batch SGD, that is, improving the worse bound for the noise variance of grafting gradient so that the performance of GGD can be improved as well.

C.3 Additional Experiments on Ridge Regression

To further check the performances of grafting gradient based methods on hard problem, that is, the ill-conditioned problem with extremely large condition number κ. We additionally run experiments to solve ridge regression problem given by

$$f_i(x) = (z_i - a_i^\top x)^2 + \frac{\lambda}{2}\|x\|^2.$$

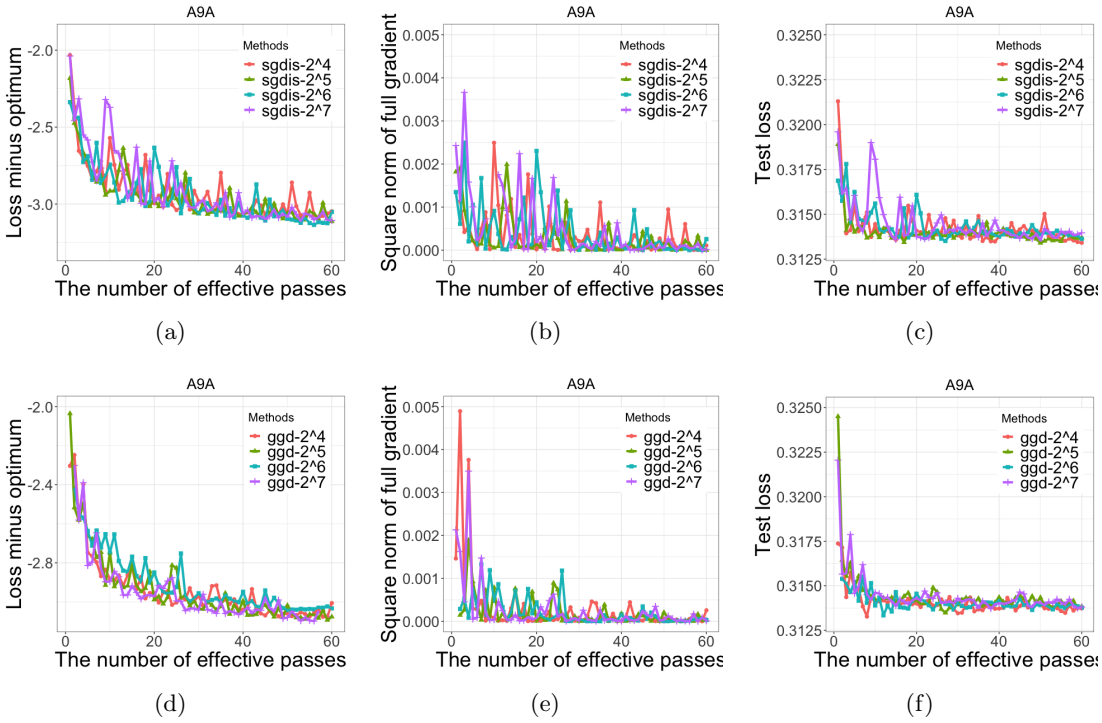


Figure 15: Comparisons of the train loss (left), square norm of full gradient (middle) and test loss (right) on a9a.

Data set	d	n (train)	Sparsity	n (test)	$\bar{L}$	$\max L$	κ
ONR	58	31,715	72.01%	7,929	0.0004	2	1.34221×10^7

Table 5: Summary of ONR data set

The objective function thus can be written in matrix form as

$$f(x) = \frac{1}{n}(z - Ax)^\top (z - Ax) + \frac{\lambda}{2} x^\top x, \tag{83}$$

where $z_{(n\times 1)}$ and $A_{(n\times d)}$ are data samples from *OnlineNewsPopularity* (*ONR* for short) data set (Fernandes and Sernadela, 2015) with z indicating the responses to be predicted and A being the data matrix. The features and targets are normalized to the interval $[0, 1]$. Since *ONR* does not have a testing set, we randomly split it into the training set and testing set with 80% for training and 20% for testing. We know that the condition number of (83) admits a closed form expression $\kappa = \lambda_{max}(H)/\lambda_{min}(H)$, where H is the hessian matrix of (83), which can be written as $H = (2/n)A^\top A + \lambda \cdot I$, and λ_{max}, λ_{min} represent the largest and smallest eigenvalue of given matrix. The penalty parameter λ is set to be 10^{-6} so that the corresponding condition number can be extremely large. We report the condition number of (83) and some other information about *ONR* data set in Table 5.

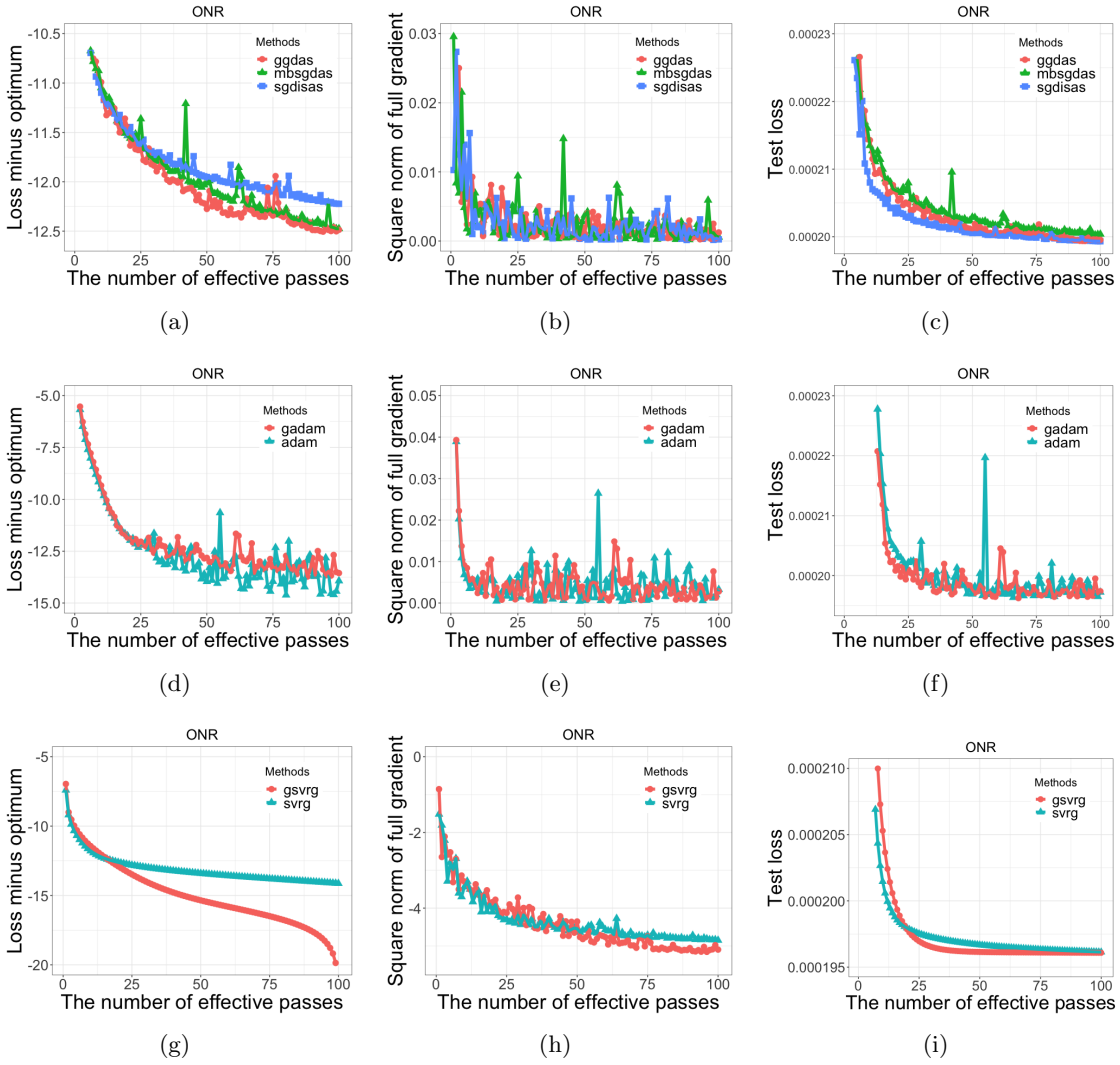


Figure 16: Comparisons of the train loss (left), square norm of full gradient (middle) and test loss (right) on ONR.

In the following experiments, we compare the performances of GGD, MBSGD and SGD-IS with diminishing stepsize sequences, the performances of Adam and GAdam and the performances of SVRG and GSVRG. For fair comparison, the subsampled set size m is set to be 32 and the batch size $b = 2$ for grafting gradient based methods. The mini-batch size of mini-batch SGD, SVRG and Adam is set to be 64 and the one-shot importance sampling probability is given by $P_{r_i} \propto L_{r_i}$. The t-inverse learning rate schedule is used for methods with diminishing stepsize sequences. For variance reduction methods, the update period is set to be n/m and for adaptive stepsize methods, the hyperparameters are set by default. The results are presented in Figure 16.

LAYER	SHAPE	OUTPUT
data layer		$1 \times 32 \times 32$
conv1-BN-ReLU	$6 \times 5 \times 5$	$6 \times 28 \times 28$
MAX-pool2	$6 \times 2 \times 2$	$6 \times 14 \times 14$
conv3-BN-ReLU	$16 \times 5 \times 5$	$16 \times 10 \times 10$
MAX-pool4	$16 \times 2 \times 2$	$16 \times 5 \times 5$
Fully-connect5-ReLU	240	120
Fully-connect6-ReLU	120	84
Fully-connect7	84	10
softmax-loss	10	10

Table 6: Complete architecture for LeNet-5

From these results, we can see that grafting gradient based methods can still achieve comparable performances even when the condition number is extremely large. Results from Figures 16(a) and 16(c) again confirm the doubly robust property possessed by GGD. The performances of MBSGD and GGD are quite similar in terms of train loss, and GGD and SGD-IS achieves a lower test loss. However, SGD-IS does not perform well in terms of train loss. In Figrue 16(a), the improvement of GGD over SGD-IS is significant. In our view, the reason behind the bad performances of SGD-IS on *ONR* data set may be over-sampling. From Table 5, $L_{max}/\bar{L} \approx 5000$ indicates that some data of great importance, i.e., with relatively large L constant, may be sampled over and over again. Hence, the training process of SGD-IS is more like minimizing the loss with respect to a small subset which contains important data samples instead of minimizing the average loss of whole training data set. In constrast, over-sampling important samples is less likely to occur in the training process of GGD as illustrated in our toy example. As for the comparison between Adam and GAdam, their performances are quite similar. For the variance reduction methods, the improvement of GSVRG over SVRG is obvious. GSVRG achieves a lower train loss and $\|\nabla f(x^k)\|^2$. Although GSVRG and SVRG achieve comparable test losses, GSVRG uses less epoches to reach that goal.

C.4 Test Loss Results and Network Architectures

Figures 17, 18 and 19 plot the test losses of previous experiments given in Section 6. From these results we can see that when minimizing the L_2-regularized logistic loss, the performances of GGD based methods and SGD based methods are quite similar except that GGD achieves a lower test loss for *covtype* and *rcv1* data set compared with vanilla SGD, and GGD with diminishing stepsize sequence achieve a lower test loss for *ijcnn* data set compared with SGD with importance sampling. When training CNNs on *MNIST* and *CIFAR-10* data sets, only SVRG achieves lower test loss than GSVRG for *MNIST* data set and Adam achieves lower test loss than GAdam for *CIFAR-10* data set.

In the end, the complete architectures for two CNNs are listed in Tables 6 and 7. We use tuple $(C \times H \times W)$ in which C represents the number of channels, H is the height in pixels and W is the width in pixels to show the shape of input data and layers of two CNNs.

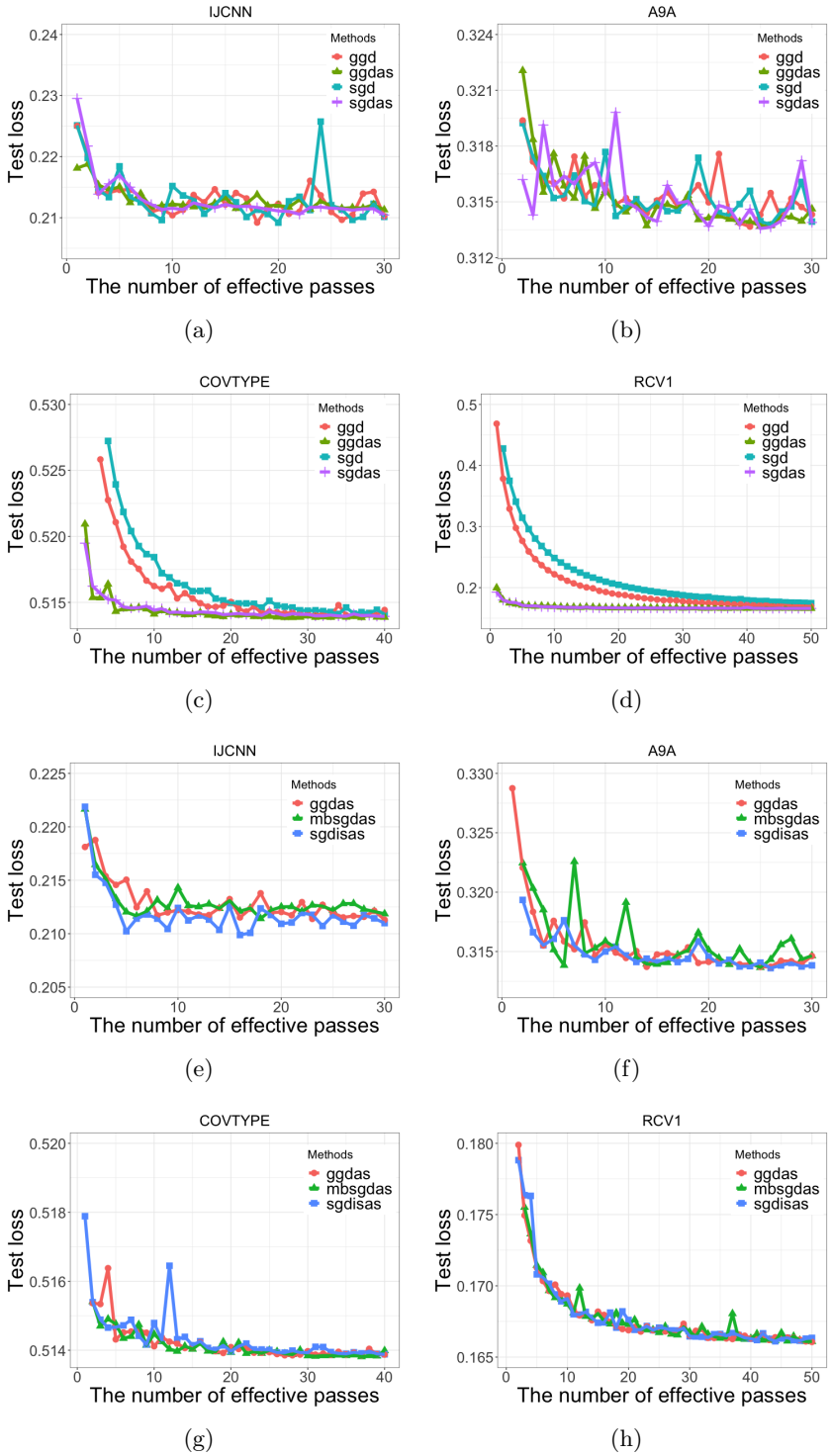


Figure 17: Comparisons of the test loss between SGD, GGD, mini-batch SGD, SGD with importance sampling methods on ijcnn1, a9a, covtype and rcv1.

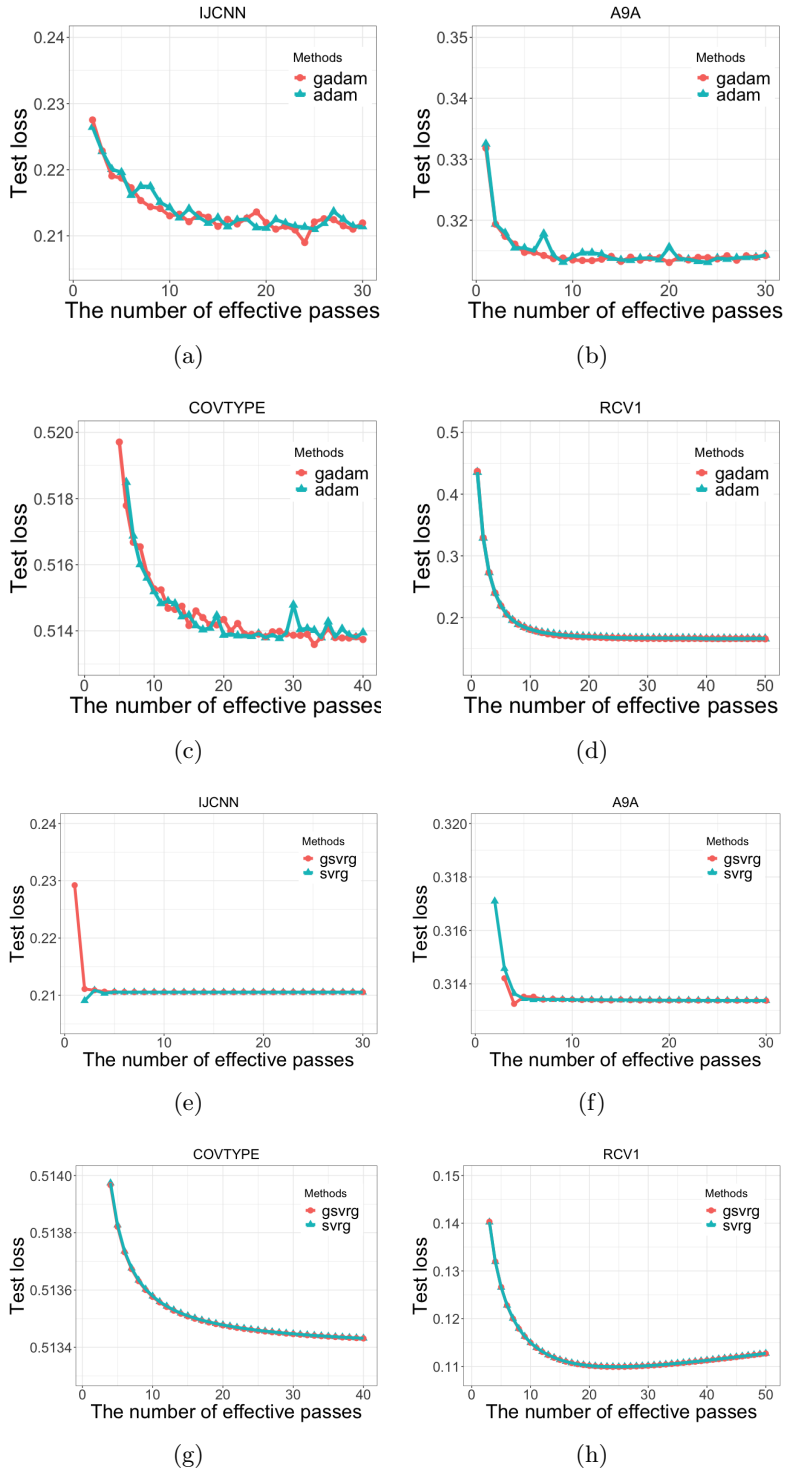


Figure 18: Comparisons of the test loss between Adam, GAdam, SVRG and GSVRG methods on ijcnn1, a9a, covtype and rcv1.

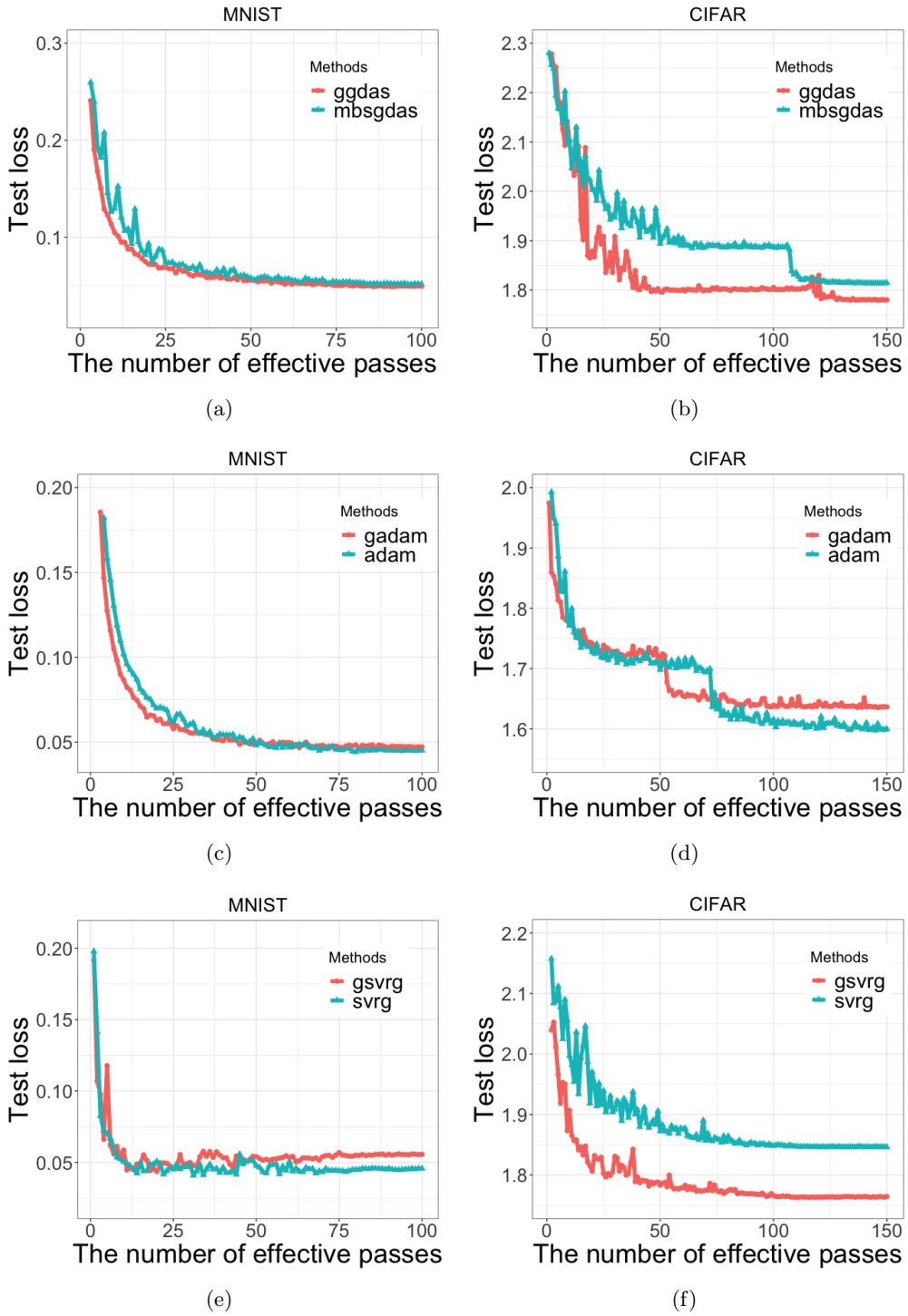


Figure 19: Comparisons of the test loss between MBSGD and GGD, Adam and GAdam, SVRG and GSVRG methods respectively on MNIST and CIFAR-10.

LAYER	SHAPE	OUTPUT
data layer		$3 \times 28 \times 28$
conv1-ReLU	$128 \times 3 \times 3$	$128 \times 28 \times 28$
conv2-ReLU	$128 \times 3 \times 3$	$128 \times 28 \times 28$
conv3-BN-ReLU	$128 \times 3 \times 3$	$128 \times 28 \times 28$
MAX-pool3	$128 \times 3 \times 3$	$128 \times 14 \times 14$
conv4-ReLU	$256 \times 3 \times 3$	$256 \times 14 \times 14$
conv5-ReLU	$256 \times 3 \times 3$	$256 \times 14 \times 14$
conv6-BN-ReLU	$256 \times 3 \times 3$	$256 \times 14 \times 14$
MAX-pool6	$256 \times 3 \times 3$	$256 \times 7 \times 7$
conv7-ReLU	$320 \times 3 \times 3$	$320 \times 5 \times 5$
conv8-ReLU	$320 \times 1 \times 1$	$320 \times 5 \times 5$
conv9-ReLU	$10 \times 1 \times 1$	$10 \times 5 \times 5$
AVE-pool9	$10 \times 5 \times 5$	$10 \times 1 \times 1$
softmax-loss	10	10

Table 7: Complete architecture for cifar10-nv

References

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *The Journal of Machine Learning Research*, 18(1):8194–8244, 2017.

Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. In *International conference on machine learning*, pages 699–707. PMLR, 2016.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

Andrew Cotter, Ohad Shamir, Nati Srebro, and Karthik Sridharan. Better mini-batch algorithms via accelerated gradient methods. *Advances in Neural Information Processing Systems*, 24, 2011.

Dominik Csiba and Peter Richtárik. Importance sampling for minibatches. *The Journal of Machine Learning Research*, 19(1):962–982, 2018.

Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

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.

Alexandre Défossez, Léon Bottou, Francis Bach, and Nicolas Usunier. A simple convergence proof of adam and adagrad. *arXiv preprint arXiv:2003.02395*, 2020.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.

Ayoub El Hanchi and David Stephens. Adaptive importance sampling for finite-sum optimization and sampling with decreasing step-sizes. *Advances in Neural Information Processing Systems*, 33:15702–15713, 2020.

Vinagre Pedro Cortez Paulo Fernandes, Kelwin and Pedro Sernadela. Online News Popularity. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C5NS3V.

Nidham Gazagnadou, Robert Gower, and Joseph Salmon. Optimal mini-batch and step sizes for saga. In *International Conference on Machine Learning*, pages 2142–2150. PMLR, 2019.

Igor Gitman and Boris Ginsburg. Comparison of batch normalization and weight normalization algorithms for the large-scale image classification. *arXiv preprint arXiv:1709.08145*, 2017.

Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. A unified theory of sgd: Variance reduction, sampling, quantization and coordinate descent. In *International Conference on Artificial Intelligence and Statistics*, pages 680–690. PMLR, 2020.

Robert M. Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. Sgd: General analysis and improved rates. In *International Conference on Machine Learning*, pages 5200–5209. PMLR, 2019.

Robert M. Gower, Mark Schmidt, Francis Bach, and Peter Richtárik. Variance-reduced methods for machine learning. *Proceedings of the IEEE*, 108(11):1968–1983, 2020.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234. PMLR, 2016.

Samuel Horváth and Peter Richtárik. Nonconvex variance reduced optimization with arbitrary sampling. In *International Conference on Machine Learning*, pages 2781–2789. PMLR, 2019.

Xinmeng Huang, Kun Yuan, Xianghui Mao, and Wotao Yin. An improved analysis and rates for variance reduction under without-replacement sampling orders. *Advances in Neural Information Processing Systems*, 34:3232–3243, 2021.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in Neural Information Processing Systems*, 26, 2013.

Tyler B Johnson and Carlos Guestrin. Training deep models faster with robust, approximate importance sampling. *Advances in Neural Information Processing Systems*, 31, 2018.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pages 2525–2534. PMLR, 2018.

Ahmed Khaled and Peter Richtárik. Better theory for sgd in the nonconvex world. *arXiv preprint arXiv:2002.03329*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jakub Konečnỳ, Jie Liu, Peter Richtárik, and Martin Takáč. Mini-batch semi-stochastic gradient descent in the proximal setting. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):242–255, 2015.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Nicolas Le Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. *Advances in Neural Information Processing Systems*, 25, 2012.

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

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Yanli Liu, Yuan Gao, and Wotao Yin. An improved analysis of stochastic gradient descent with momentum. *Advances in Neural Information Processing Systems*, 33:18261–18271, 2020.

Grigory Malinovsky, Alibek Sailanbayev, and Peter Richtárik. Random reshuffling with variance reduction: New analysis and better rates. *arXiv preprint arXiv:2104.09342*, 2021.

Konstantin Mishchenko, Ahmed Khaled, and Peter Richtárik. Random reshuffling: Simple analysis with vast improvements. *Advances in Neural Information Processing Systems*, 33:17309–17320, 2020.

Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural importance sampling. *ACM Transactions on Graphics (ToG)*, 38(5):1–19, 2019.

Deanna Needell, Rachel Ward, and Nati Srebro. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. *Advances in Neural Information Processing Systems*, 27, 2014.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

Lam M. Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*, pages 2613–2621. PMLR, 2017.

Michael JD Powell. On search directions for minimization algorithms. *Mathematical programming*, 4:193–201, 1973.

Xun Qian, Zheng Qu, and Peter Richtárik. Saga with arbitrary sampling. In *International Conference on Machine Learning*, pages 5190–5199. PMLR, 2019.

Sashank J. Reddi, Ahmed Hefny, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International Conference on Machine Learning*, pages 314–323. PMLR, 2016.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1):83–112, 2017.

Othmane Sebbouh, Nidham Gazagnadou, Samy Jelassi, Francis Bach, and Robert Gower. Towards closing the gap between the theory and practice of svrg. *Advances in neural information processing systems*, 32, 2019.

Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical Programming*, 127(1):3–30, 2011.

Fanhua Shang, Kaiwen Zhou, Hongying Liu, James Cheng, Ivor W. Tsang, Lijun Zhang, Dacheng Tao, and Licheng Jiao. Vr-sgd: A simple stochastic variance reduction method for machine learning. *IEEE Transactions on Knowledge and Data Engineering*, 32(1): 188–202, 2018.

Tijmen Tieleman, Geoffrey Hinton, et al. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

Phuong Thi Tran et al. On the convergence proof of amsgrad and a new version. *IEEE Access*, 7:61706–61716, 2019.

Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. *The Journal of Machine Learning Research*, 21(1):9047–9076, 2020.

Zhuang Yang, Zengping Chen, and Cheng Wang. Accelerating mini-batch sarah by step size rules. *Information Sciences*, 558:157–173, 2021.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018.

Matthew D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *International Conference on Machine Learning*, pages 1–9. PMLR, 2015.

Kaiwen Zhou, Fanhua Shang, and James Cheng. A simple stochastic variance reduced algorithm with fast convergence rates. In *International Conference on Machine Learning*, pages 5980–5989. PMLR, 2018.

Fangyu Zou, Li Shen, Zequn Jie, Weizhong Zhang, and Wei Liu. A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11127–11135, 2019.