On Convergence of Incremental Gradient for Non-Convex Smooth Functions

Anastasia Koloskova [1] **Nikita Doikov** [1] **Sebastian U. Stich** [2] **Martin Jaggi** [1]

Abstract

In machine learning and neural network optimization, algorithms like incremental gradient, and shuffle SGD are popular due to minimizing the number of cache misses and good practical convergence behavior. However, their optimization properties in theory, especially for non-convex smooth functions, remain incompletely explored.

This paper delves into the convergence properties of SGD algorithms with arbitrary data ordering, within a broad framework for non-convex smooth functions. Our findings show enhanced convergence guarantees for incremental gradient and single shuffle SGD. Particularly if n is the training set size, we improve n times the optimization term of convergence guarantee to reach accuracy ε from $\mathcal{O}\left(n/\varepsilon\right)$ to $\mathcal{O}\left(1/\varepsilon\right)$.

1. Introduction

In this paper we study the problem of minimizing the finite-sum objective:

$$\min_{\mathbf{x}\in\mathbb{R}^d}\left[f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x})\right]. \tag{1}$$

We denote by n the number of functions $f_i\colon \mathbb{R}^d \to \mathbb{R}$. Every function f_i is assumed to be smooth and can be non-convex. This optimization problem arises in many practical applications. For example, in the training of machine learning models with n being the training set size and each $f_i(\mathbf{x})$ being the loss of the model on the i-th datapoint, while $\mathbf{x}$ is the vector of the model parameters.

A common method for solving optimization problems of the form (1) is the Stochastic Gradient Descent (SGD) algorithm and its various modifications (Lan, 2020). During each iteration $t \geq 0$ of the method, an index $i_t \in [n] := \{1, \dots, n\}$ (or a subset of indices) is selected upon which a gradient step is performed:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma \nabla f_{i_t}(\mathbf{x}_t), \tag{2}$$

where $\gamma > 0$ is a step size. Defining how the index i_t is selected in (2) is crucial for the performance of this algorithm. The standard theoretical analysis often assumes that the index $i_t \sim [n]$ is chosen uniformly at random (a variant we refer to as SGD in this paper). However, in practice, a specific order of selecting the gradients can be *predefined* by a configuration of the system. For example, many standard frameworks use dataloaders that process data in epochs. Within each epoch, we access all indices in a permutation. The permutation can vary across epochs, or fixed at once in the beginning. The latter is one of the most popular approaches in practice due to simplicity of its implementation and high efficacy of memory usage, minimizing the number of cache misses. From the optimization perspective, this means that after performing the gradient step (2) for a certain index i_t, the next time we access the function f_{i_t} again is *exactly after n iterations*. Despite being widely approved by the technical and engineering needs, these variations of SGD present one of the most challenging scenario for the analysis, because of mutual dependencies between consecutive steps.

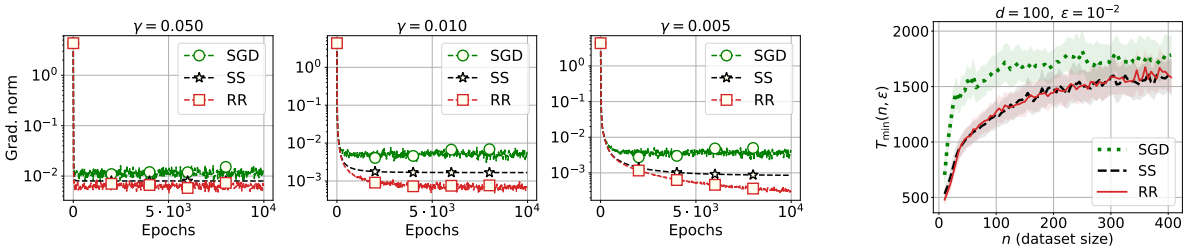
In this paper, our goal is to understand the convergence properties of algorithm (2) with a predefined selection of indices shared within one epoch. An important particular case is called the *incremental gradient descent* (IG) algorithm. In recent decades, it has received significant attention in the optimization literature (Bertsekas, 2011). The basic version of this method simply substitutes

$$i_t \quad := \quad t \bmod n + 1 \tag{3}$$

into algorithm (2). Note that when the initial order of functions in (1) is randomly permuted once in the beginning, this selection rule is equivalent to the so-called Single Shuffle (SS) method, as opposed to Random Reshuffle (RR), which permutes the functions after every epoch. Even when putting aside cache efficiency in practical implementations, both SS and RR show good practical behavior often surpassing classic SGD with uniform sampling (see e.g. Fig. 1). This makes them popular in the machine learning and optimization community (Mishchenko et al., 2020; Lu et al., 2022b) and very often these are the methods of choice that

[1]Machine Learning and Optimization Laboratory (MLO), EPFL, Lausanne, Switzerland [2]CISPA Helmholtz Center for Information Security, Saarbrücken, Germany. Correspondence to: Anastasia Koloskova <anastasia.koloskova@epfl.ch>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.



(a) Convergence of the gradient norm with epochs ($d = 100, n = 50$)

(b) Number of iterations to reach ε accuracy (the lower is the better).

Figure 1. Minimizing stochastic quadratic function for different strategies of sampling the gradients. Random Reshuffling (RR) and Single Shuffling (SS) work better than SGD.

are implemented in practical systems. Another important example of a significant benefit of the incremental algorithms is second-order optimization. Thus, it was shown in (Rodomanov & Kropotov, 2016) that the cyclic rule (3) for updating a second-order model in the Newton methods is essential to preserve the local superlinear convergence.

Despite their popularity and wide practical approval, the convergence rates for the methods with arbitrary data orders and, specifically, for the incremental gradients (3) are not well understood. Up to our knowledge, the best results for IG show a dependency on the dataset size n of the form $\Omega\left(\frac{n}{\varepsilon}\right)$, i.e. a linear dependence on n (Mishchenko et al., 2020; Lu et al., 2022b). This means that n-iterations of IG give comparable progress to one step of the (full-batch) gradient method. This is in contrast to convergence rates of SGD, for which the optimization term is $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ and does not depend on n. This is also in contrast to empirical observations.

In this work, we provide a novel analysis for a general family of algorithms of the form (2), with an emphasis on the incremental gradient methods and establish new convergence rates for non-convex optimization that are significantly better than the previously known ones (see Table 1), thus advancing in a long-standing problem in optimization theory.

In order to achieve such an improvement, we develop a new proof technique, drawing inspiration from previous work on SGD with linearly correlated noise, such as occurring in optimization with differential privacy (Koloskova et al., 2023). In particular, we divide all iterations $t = 0, \ldots, T$ of the method into smaller chunks of size $\tau = \Theta\left(1/L\gamma\right)$ with L being the smoothness constant and γ the step size, and analyze each of the chunks separately. This is different from the previous works as they usually consider the correlation periods of a fixed size n, regardless of how large n is.

We can summarize our contributions as follows:

- We derive convergence rates for SGD with *arbitrary* data orderings for non-convex smooth functions, that cover a wide range of algorithms, including, but not restricted to Random Reshuffling (RR), Single Shuffle (SS) and Incremental Gradient (IG).
- Our convergence rates strictly improve all the rates in prior works for the non-convex smooth functions in the case of the Incremented Gradient and Single Shuffle (see Table 1).
- Our theoretical analysis technique is novel and of independent interest in comparing other schemes.

2. Related work

In this section, we review some of the most well-established practical and theoretical results on incremental gradient, shuffle SGD variants and SGD with other data orders that were known prior to our work.

Practical observations. Preceding any theoretical findings, it was initially empirically discovered that the behaviour of stochastic methods can vary significantly depending on the order of stochastic gradients. It has been widely reported in the literature that random shuffling or cycling through the data over a fixed shuffle (permutation) yields better convergence than sampling the datapoints uniformly at random. For instance, (Nedic & Bertsekas, 2001) discovered this in the context of subgradient methods, while (Bottou, 2009) and (Shalev-Shwartz et al., 2011) were among the first to discuss this observation in the context of machine learning problems. They posed an open problem to justify this theoretically. (Recht & Ré, 2011) showed that for the matrix completion problem, random shuffling can be several orders of magnitude faster. Similarly, in the context of neural network training, it was recommended by (Bengio, 2012) to shuffle the dataset once and then use a fixed order of the gradients. Currently, random reshuffling is standard practice in training deep learning models (Paszke et al., 2019; Sun, 2020). It is also used routinely for training large language models (Chowdhery et al., 2022; Touvron et al., 2023), with a small number of epochs over the randomly shuffled training data. Shuffling methods were predated by the incremental gradient methods that pass over the data in a given order (Kohonen, 1974; Luo, 1991; Grippo, 1994; Bertsekas, 2015). Studies on the effects of data ordering

Algorithm	Prior works	This work
Incremental Gradient / Single Shuffle	$LF_0 \cdot \mathcal{O}\left(\frac{n}{\varepsilon} + \frac{n\sigma_{\text{SGD}}}{\varepsilon^{3/2}}\right)$	$LF_0 \cdot \mathcal{O}\left(\frac{1}{\varepsilon} + \mathbf{min}\left\{\frac{n\sigma_{\text{SGD}}}{\varepsilon^{3/2}}, \frac{n\sigma^2_{\text{SGD}}}{\varepsilon^2}\right\}\right)$

Table 1. Prior best known (Mohtashami et al., 2022; Lu et al., 2022b; Mishchenko et al., 2020) complexity (number of iterations to achieve accuracy ε) for the special case of Incremental Gradient and Single Shuffle methods covered in our framework, compared to the improved rate derived in our work. $\mathcal{O}(\cdot)$ hides an absolute numerical constant. Note that the standard complexity of SGD is $LF_0 \cdot \mathcal{O}(\frac{1}{\varepsilon} + \frac{\sigma^2_{\text{SGD}}}{\varepsilon^2})$, which is also covered by our analysis as a special case (7). However, the practical behaviour of Single Shuffle is usually better than that of SGD (see Fig. 1 and Section 6 with our numerical experiments).

on neural network training can be traced back at least to the 1960s (Widrow & Hoff, 1960).

Theoretical guarantees. There were many attempts and breakthroughs to theoretically explain the good empirical behaviour of the gradient methods with reshuffling for different problem classes. Convergence rates of the incremental gradient methods with random reshuffling for *convex* optimization are dated back to (Nedić & Bertsekas, 2001) (see also (Bertsekas, 2011)). Over the recent years, significant attention has been dedicated to the *strongly convex* case. A series of works could establish that RR can converge faster than SGD. Among the first analyses is (Recht & Ré, 2012) that focuses on quadratic least squares problem. Follow-up work also focuses on quadratics or relaxes the assumption by imposing second-order smoothness assumptions. Gürbüzbalaban et al. (2015); Haochen & Sra (2019); Safran & Shamir (2020) provided the lower bounds for RR and SS for strongly convex functions. Subsequently, (Rajput et al., 2020) showed how to match the lower bound for RR when the function f is quadratic and after a large enough number of epochs. For linear regression Yun et al. (2021b) show that SS is better than RR and better than SGD. The second-order smoothness assumption could be dropped in (Jain et al., 2019; Safran & Shamir, 2021) and (Mishchenko et al., 2020). The latter makes the observation that introducing a specific notion of variance that takes the random permutation into account could facilitate the analysis in the convex and strongly convex setting. The results were extended to the non-convex setting under the PL condition (Ahn et al., 2020; Nguyen et al., 2020) and for the general *non-convex smooth* setting in (Lu et al., 2022b; Mishchenko et al., 2020; Mohtashami et al., 2022). Tighter lower complexity bounds for strongly convex functions and functions satisfying the PL condition were established recently in (Cha et al., 2023). In the last years, a significant attention was paid to the distributed and federated stochastic methods with random reshuffling (see (Yun et al., 2021a; Sadiev et al., 2022; Malinovsky et al., 2022; Mishchenko et al., 2022; Cho et al., 2023) and references therein), which are important for training large-scale and decentralized models.

Moreover, all of the convergence rates for non-convex functions require the number of epochs to be unreasonably large for random reshuffling to be better than SGD.

Arbitrary orderings. (Lu et al., 2022b; Mohtashami et al., 2022) analysed arbitrary data orderings in SGD (not restricted to permutations), including shuffle SGD and proposed an algorithm of greedy data selection to select data orders that would lead to the fastest convergence, based on their analysis. (Lu et al., 2022a) proposed a more practical (than greedy order) data order selection that is faster than random shuffle.

There exist many more applications where different data orderings appear naturally, such as in Markovian SGD (Johansson et al., 2009; Duchi et al., 2011; Even, 2023) or certain federated learning scenarios (Eichner et al., 2019). Yun et al. (2022) analyse local SGD for distributed learning when every node applies random reshuffling. Using a provably faster permutation-based example ordering in distributed training was recently studied in (Feder Cooper et al., 2023).

3. The Algorithm

Recall that we study the finite-sum minimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left[f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x})\right] \tag{1}$$

where n denotes the number of functions.

While we are mainly interested in analysing convergence properties of the incremental gradient (IG) algorithm, we will study it under the more general framework that allows for *arbitrary* data ordering:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma \nabla f_{i_t}(\mathbf{x}_t) \tag{2}$$

where $t = 0, \ldots, T$ and $i_t \in [n] := \{1, \ldots, n\}$ denotes the index of the datapoint chosen at iteration t.

In our work we allow for any possible strategies of choosing datapoints orders, i.e. sequences of indices $(i_t)_{0\le t\le T} := (i_0, \ldots, i_t, \ldots, i_T)$—deterministic or random.

The results that we present in this work can be straightforwardly extended to the mini-batch setting, where in each iteration a mini-batch gradient is computed on a subset of the data. In this case, we can denote by I_t a set of indices taken in iteration t and work with

$f_{I_t}(\mathbf{x}) := \frac{1}{|I_t|}\sum_{j\in I_t} f_j(\mathbf{x})$. However, for simplicity of the presentation, we will mainly focus on the iterations of the form (2), and i_t being just one index.

Let us list some examples of the popular algorithms covered in our framework (2):

Example 3.1 (SGD)**.** The classic SGD algorithm (Robbins & Monro, 1951) samples in each iteration t the next data sample uniformly at random from $[n]$, i.e. $i_t \sim [n]$. This means that in one epoch (n iterations), some samples may be selected more than once and others may be missing (sampling with replacement). Most theoretic analyses of stochastic methods (see, e.g. the book by (Lan, 2020)) are based on this type of sampling, since independently choosing a data point in each iteration significantly facilitates the proofs.

Example 3.2 (Incremental Gradient (IG))**.** The incremental gradient method passes over the data samples in cycles (which we will also refer to as epochs), i.e. $i_t = t \bmod n + 1$. This method has been among the earliest used for training neural networks (Kohonen, 1974; Luo, 1991; Bertsekas, 2011) but comes with unfavorable worst-case convergence guarantees (Gürbüzbalaban et al., 2019).

Example 3.3 (Single Shuffle (SS))**.** Single shuffle SGD also passes over the data sample in cycles, but instead of using the predefined order, the data order is determined by a randomly chosen permutation π of the index set $[n]$, and we set $i_t = \pi(t \bmod n + 1)$. This variant is also sometimes used in practical implementations (Bengio, 2012).

Example 3.4 (Random Reshuffling (RR))**.** SGD with random reshuffling (RR) (Nedić & Bertsekas, 2001) is a further popular variant that passes in cycles of length n over the data. The main difference compared to SS is that in RR a *new* permutation π_k is drawn uniformly at random at the beginning of each epoch k, and then $i_t = \pi_k(t \bmod n+1)$, which is also called *sampling without replacement*. This variant is frequently a default option in training modern neural networks (Goyal et al., 2017; Paszke et al., 2019).

The following example is less conventional in the literature, it serves to highlight that our framework also allows orderings that completely disregard parts of the data, leading to biased training.

Example 3.5 (Single Function)**.** Since our algorithm (2) allows for any orders of the data, it is also allowed to always sample the same (for example the first) function, i.e. $i_t \equiv 1\ \forall t$. In this case, we cannot expect the convergence to the exact optima of f in (1), but to the minimizer of f_1. We will see that in this case, our theory can quantify the neighbourhood size to which the algorithm converges. While this example is only illustrative, it serves as a proxy to a more realistic scenario when some functions might be missing due to e.g. being distributed over the nodes/devices that are offline.

Any other deterministic or randomized orders are also possible, such as Markovian SGD (Johansson et al., 2009; Even, 2023).

Our analysis will cover arbitrary orders and include all of the examples from Examples 3.1 to 3.5. However, it is important to note that not all examples will demonstrate improved convergence under our analysis. The primary focus of our study will be on the Incremental Gradient and Single Shuffle SGD algorithms (Examples 3.2, 3.3), where we notably improve existing convergence guarantees.

4. Assumptions

For our theoretical analysis, we will use the following standard assumption on the smoothness of functions (see, e.g. (Nesterov, 2018)).

Assumption 4.1. Each of f_i is L-smooth, i.e. $\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \le L\|\mathbf{x}-\mathbf{y}\|\ ,\ \ \forall \mathbf{x},\mathbf{y}\in\mathbb{R}^d$.

4.1. Quantifying the data orders

Different data orders $(i_0,\dots,i_T)$ in Algorithm (2) lead to the different practical performance and the different convergence rates. Consider the two examples: (i) i_t is chosen uniformly at random from $[n]$, corresponding to the SGD Example 3.1 and (ii) i_t is always chosen to be the first function $i_t \equiv 1$, as in Example 3.5. Clearly while SGD can effectively minimize the original objective function f defined in (1), the second algorithm converges to the minimum of f_1, and might not converge to the minima of f in general, if $f_1 \neq f$.

In order to provide a unified analysis that captures the convergence of arbitrary data ordering in Algorithm (2), we need to introduce a quantity that measures how does the datapoint order $(i_0,\dots,i_T)$ affect the convergence rate. In this work, we propose to use the following quantity

Definition 4.2 (Sequence correlation)**.** Let $(i_0,\dots,i_T)$ denote a (possibly random) sequence of indices $i_t \in [n], t = 0,\dots,T$. For a given $\tau \ge 1$ (that we call *effective correlation time*), we divide the full sequence into $\lfloor \frac{T}{\tau} \rfloor$ consecutive chunks of size τ. We call these chunks as *correlated periods*. We further define for every period $k = 0,\dots,\lfloor \frac{T}{\tau} \rfloor$, the sequence variance as

$$\sup_{\mathbf{x}\in\mathbb{R}^d} \max_{\substack{k=0,\dots,\lfloor \frac{T}{\tau} \rfloor \\ j=0,\dots,\tau-1}} \mathbb{E}\left[\phi_{k\tau+j}(\mathbf{x}) \,\middle|\, i_0,\dots,i_{k\tau-1}\right] \le \sigma_\tau^2 \quad (4)$$

where

$$\phi_{k\tau+j}(\mathbf{x}) = \left\| \sum_{t=k\tau}^{\min\{k\tau+j,T\}} (\nabla f_{i_t}(\mathbf{x}) - \nabla f(\mathbf{x})) \right\|^2$$

where the expectation is taken over the choice of the random sequence $(i_t)_{k\tau\le t\le T}$ conditioning on the past sequence

order $(i_t)_{t<k\tau}$. We note that σ_τ^2 depends on the distribution from which the sequence $(i_t)_{0\le t\le T}$ is drawn, but we omit this dependence in the text when it is clear from the context.

For the special case of the correlated periods of size one, i.e. $\tau = 1$ and indices sampled uniformly at random, $i_t \sim [n]$, this definition recovers the standard SGD variance as used in (5) below. When considering $\tau > 1$, this measure can capture the joint effect of sequence order and the gradients of the individual functions. This measure shows how close the average of prefix of individual gradients $\nabla f_{i_t}(\mathbf{x})$ stays to the full gradient $\nabla f(\mathbf{x})$ during the period of τ consecutive steps.

4.2. Comparison to the prior quantities

We will now explore how the sequence correlation quantity we introduced earlier, as defined in Definition 4.2, relates to similar concepts in existing literature. Additionally, we will discuss the motivations behind proposing this new sequence correlation approach.

We first start with comparing our sequence correlation with classic variance assumptions that appear in the analysis of stochastic gradient methods (Bubeck, 2015; Lan, 2020).

Classic bounded variance assumption. One of the most common assumptions used in analysing SGD-type algorithms (Lan, 2012; Dekel et al., 2012) is bounded variance of the gradients. They assume that σ^2_{SGD} is finite, where σ^2_{SGD} is defined as

$$\sigma^2_{\text{SGD}} := \sup_{\mathbf{x}\in\mathbb{R}^d} \mathbb{E}_i \left\|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\right\|^2 , \qquad (5)$$

and the expectation is taken over the uniform choice of the index $i \sim [n]$ (note that mini-batches are also allowed, effectively dividing this variance quantity by the batch size).

While this measure σ^2_{SGD} is good in quantifying the behaviour of SGD when the datapoints are sampled with replacement, it weighs every function i uniformly. Thus it cannot successfully capture the effect of different orders in (2) on the optimization trajectory, when the functions are sampled non-uniformly.

Example 4.3 (Example when (5) fails, (Mohtashami et al., 2022))**.** As an example, consider n univariate functions ($x \in \mathbb{R}$) defined as $f_i(x) = \frac{1}{2}(x-1)^2$, for the first half: $1 \le i \le \frac{n}{2}$, and for the second half we define $f_i(x) = \frac{1}{2}(x+1)^2$, $\frac{n}{2} < i \le n$, with n being an even number. Let us consider two instances of algorithm (2). In the first instance, the method cycles through functions in the initial order $f_1, \dots, f_n$ without shuffling, and in the second instance it cycles in the following order: $f_1, f_{\frac{n}{2}+1}, f_2, f_{\frac{n}{2}+2}, \dots, f_n$. In the second case the algorithm will converge faster than in the first one. However, by assuming only smoothness (Assumption 4.1) and the classic boundedness of variance (5), there is no way to distinguish the two cases, as these assumptions do not hold any information about the chosen order.

This example motivates the need to couple the variance assumption in (5) with the order of the gradients used in the method.

Variance assumptions that take the data-ordering into account. Several recent works (Lu et al., 2022b; Mohtashami et al., 2022; Mishchenko et al., 2020) proposed assumptions that depend on the data order.

Most prior works considered correlation periods of a fixed size $\tau = n$. For instance, (Mohtashami et al., 2022) define a finite constant σ^2_{EPOCH} (assuming it is finite $\sigma^2_{\text{EPOCH}} < \infty$), such that

$$\sigma^2_{\text{EPOCH}} := \sigma^2_{\text{n}} . \qquad (6)$$

A very similar assumption was used by (Lu et al., 2022b) where they allow the staring point to be an arbitrary index, but the correlation distance τ can be as large as T. We note that both these prior works impose an additional assumption that considered data orderings that do not deviate much from the full gradient, while we do not impose such conditions. Both of the works considered a refined growth condition, which we do not consider in this work for simplicity.

Let us explain how an epoch based assumption $\sigma^2_{\text{EPOCH}} < \infty$ can help in characterising the order of the gradients as compared to the classic variance σ^2_{SGD}.

Example 4.4. Indeed, in Example 4.3 above we could not distinguish the two different sequences using only the variance σ^2_{SGD}. However, for the first sequence, we can use a bound

$$\sigma^2_{\text{EPOCH}}\big[(1,2,\dots,n)\big] \quad = \quad \tfrac{n^2}{8}\,\sigma^2_{\text{SGD}},$$

while in the second case, we can set

$$\sigma^2_{\text{EPOCH}}\big[(1,\tfrac{n}{2}+1,2,\tfrac{n}{2}+2,\dots,n)\big] \quad = \quad \tfrac{1}{2}\,\sigma^2_{\text{SGD}}\,.$$

Hence, employing the global bound (6) on our sequence variance $\sigma^2_{k,\tau}$ for $\tau = n$, we can clearly quantify the effect of using different orders of the gradients. As we explain in the next example, it still has some limitations.

Example 4.5 (Example when Assumption (6) fails)**.** Consider a *single function* algorithm described in Example 3.5, i.e. $i_t \equiv 1\ \forall t$, then the variance of any function $i \neq 1$ is irrelevant, as the algorithm never sees them. For this sequence, let us define $\sigma^2_{\text{ONE}} = \sigma^2_{k,1}$ (for an arbitrary index $k \le T$). We can observe $\sigma^2_{k,n} = n^2\,\sigma^2_{\text{ONE}}$. However, the number of functions n should not matter for convergence properties of this algorithm, as the algorithm sees only one function.

Algorithm	Upper bound on σ_τ^2
SGD, Ex. 3.1	$\tau\,\sigma_{\text{SGD}}^2$
RR, Ex 3.4	$4\min\{\tau, n\}n\,\sigma_{\text{SGD}}^2$
IG and SS, Ex. 3.2, Ex 3.3	$\min\{\tau, n\}n\,\sigma_{\text{SGD}}^2$
Single function, Ex. 3.5	$\tau^2\,\sigma_{\text{ONE}}^2$

Table 2. Upper bounds on σ_τ^2 for notable special cases.

4.3. Our observation on effective correlation time

However, the main limitation of the prior work and of considering the epoch-based variance σ_{EPOCH}^2 is not because it cannot capture the convergence behaviour of a single function Example 3.5. It is because of its limitations when analysing *incremental gradient methods*—methods of the main interest in this work.

Due to the technical reasons, analysing theoretical convergence of (2) using the variance σ_τ^2 with the correlation period τ puts a constraint on the choice of stepsize as $\gamma < \mathcal{O}\left(1/L\tau\right)$.

Setting the correlation period $\tau = n$, as the prior works did (Lu et al., 2022b; Mohtashami et al., 2022; Mishchenko et al., 2020), therefore limits the stepsize as $\gamma < \mathcal{O}\left(1/Ln\right)$. The small stepsize means small progress at every iteration of algorithm (2), slowing down its convergence behaviour, especially when n is large. Mathematically this results in the n times slow down in the first term of convergence: $\mathcal{O}\left(n/\varepsilon\right)$ term in the convergence rate (see Table 1).

In this work, we avoid such a restriction of the stepsize by adaptively choosing the correlation period τ instead of fixing it to n. In our work, the correlation period τ is not chosen in advance, but it adapts to the stepsize γ and the smoothness constant L as $\tau = \Theta\left(1/L\gamma\right)$ so that in the end the stepsize γ becomes unrestricted by τ, but instead τ becomes restricted by the stepsize γ. This allows us to improve n times the first term of convergence to $\mathcal{O}\left(1/\varepsilon\right)$ (see Table 1).

Intuitive informal explanation. Intuitively, effective correlation time τ should capture how many iterations one needs to perform to move "sufficiently far" from the initial point. This measure should depend only on the stepsize γ and the smoothness constant L of functions f_i. Indeed, for the larger stepsizes, every step of Algorithm (2) will make the larger progress, and the iterates $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ will be further apart, reducing the effective correlation time. Similarly, the larger the smoothness constant L, the faster the function f changes, and the smaller the distance one need to have to make the iterations uncorrelated for a given f (decreasing effective correlation time). Therefore, $\tau = \Theta\left(1/L\gamma\right)$.

5. Main theorem

In this section we present our main theoretical result.

Theorem 5.1. *Let each of the functions f_i be L-smooth (Assumption 4.1). Let the stepsize γ in Algorithm (2) satisfy: $\gamma \leq \frac{1}{8\sqrt{3}L}$. Let $\tau = \left\lfloor \frac{1}{8\sqrt{3}L\gamma} \right\rfloor$, and assume that σ_τ^2 from Def. 4.2 is finite, for $k = 0, \ldots, \left\lfloor \frac{T}{\tau} \right\rfloor$, where T is a number of iterations. Then, Algorithm (2) converges at the rate:*

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}\left(\frac{F_0}{\gamma T} + L^2\gamma^2\,\sigma_\tau^2\right),$$

where $F_0 = f(\mathbf{x}_0) - f^\star$.

We give complete proof of this theorem in the Appendix.

The convergence rate consists of two terms. The first term is the optimization term, and it recovers the tight convergence rate of noiseless gradient descent ($\sigma_\tau^2 = 0$). The second term is the most interesting one, it shows the effect of chosen data orders in Algorithm (2). The convergence rate depends on the sequence variance σ_τ^2.

We note that since σ_τ^2 depends on τ, which in itself depends on the stepsize γ, we cannot directly tune the stepsize in the upper bound. However, in several notable special cases we show how one can we tune the stepsize by estimating σ_τ^2 (see Table 2), and get the final convergence rate.

5.1. Implications for the Special Cases

SGD, Ex. 3.1 Since $\sigma_\tau^2 \leq \tau\,\sigma_{\text{SGD}}^2$ (see Table 2), and using that $\tau = \Theta\left(1/L\gamma\right)$ the convergence rate in Theorem 5.1 converts to

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}\left(\frac{F_0}{\gamma T} + L\gamma\,\sigma_{\text{SGD}}^2\right), \tag{7}$$

with $\gamma \leq \frac{1}{8\sqrt{3}L}$. This recovers classical convergence rate of SGD for non-convex functions (up to constants). In particular, after tuning the learning rate (see details in Appendix E), we get the convergence of $\mathcal{O}\left(LF_0/T + \sqrt{LF_0\,\sigma_{\text{SGD}}^2/T}\right)$. This recovers classical convergence rate of SGD for non-convex functions (up to constants). In particular, after tuning the learning rate (see details in Appendix E), we get the convergence of $\mathcal{O}\left(LF_0/T + \sqrt{LF_0\,\sigma_{\text{SGD}}^2/T}\right)$.

Importantly, the previous works (Mohtashami et al., 2022; Lu et al., 2022b) on analysing arbitrary ordered sequences in SGD could not recover the tight convergence rates in these cases.

Incremental Gradient and Single Shuffle, Ex. 3.2, Ex. 3.3 Substituting $\sigma_\tau^2 \leq \min\{\tau, n\}n\,\sigma_{\text{SGD}}^2$, we get that $\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2$ is smaller than

$$\mathcal{O}\left(\frac{F_0}{\gamma T} + L\gamma\min\{1, L\gamma n\}n\,\sigma_{\text{SGD}}^2\right), \tag{8}$$

with $\gamma \leq \frac{1}{8\sqrt{3}L}$. Thus, after the stepsize tuning (see details in Appendix E), we get the rate of

$\mathcal{O}\left(LF_0/T + \min\left\{\left(LF_0 n\sigma_{\text{SGD}}/T\right)^{\frac{2}{3}}, \sqrt{LF_0 n\,\sigma^2_{\text{SGD}}/T}\right\}\right)$ which is strictly better for both of the terms than the previously best known bound $\mathcal{O}\left(LF_0 n/T + \left(LF_0 n\sigma_{\text{SGD}}/T\right)^{\frac{2}{3}}\right)$ (Mohtashami et al., 2022; Lu et al., 2022b; Mishchenko et al., 2020).

Single Function, Ex. 3.5 We now show that our convergence rate can give tight guarantees when the optimization is always performed on a single function $i_t \equiv 1$. As discussed above, in this case we have $\sigma_\tau^2 = \tau^2\sigma_1^2$, and thus we can guarantee the convergence of

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \;\leq\; \mathcal{O}\big(\tfrac{F_0}{\gamma T} + \sigma^2_{\text{ONE}}\big)\,, \qquad (9)$$

with $\gamma \leq \frac{1}{12L}$. It provides the method with convergence only to the neighbourhood of size σ^2_{ONE}, which cannot be reduced. This is expected, as the algorithm has no way to learn information about the full function f (Ajalloeian & Stich, 2020). The same convergence rate could be simply obtained by noticing that $\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq 2\left\|\nabla f_1(\mathbf{x}_t)\right\|^2 + 2\left\|\nabla f(\mathbf{x}_t) - \nabla f_1(\mathbf{x}_t)\right\|^2 \leq 2\left\|\nabla f_1(\mathbf{x}_t)\right\|^2 + 2\,\sigma^2_{\text{ONE}}$ for all $\mathbf{x}_t$ and that $\frac{1}{T}\sum_{t=0}^{T}\left\|\nabla f_1(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}\left(F_0/\gamma T\right)$ as the algorithm performs a full gradient descent on function f_1 that is L-smooth. In the Appendix we show that this convergence rate is in fact tight.

This example serves to highlight that our Theorem 5.1 can capture tightly the cases even when the selected orders i_t result in the gradients that are always far from the true gradient $\nabla f(\mathbf{x})$.

5.2. Random Reshuffling

Algorithm (2) and our Theorem 5.1 cover arbitrary orderings of the data, including random reshuffling (RR) algorithm that cycles through the dataset, reshuffling the order of the data after every epoch (Ex. 3.4).

Applying the bound on σ_τ^2 from Table 2, and tuning the stepsize γ the convergence rate in Theorem 5.1 (see details in Appendix E) converts to

$$\mathcal{O}\left(LF_0/T + \min\left\{\left(LF_0 n\sigma_{\text{SGD}}/T\right)^{\frac{2}{3}}, \sqrt{LF_0 n\,\sigma^2_{\text{SGD}}/T}\right\}\right)$$

The prior best known convergence rate of random reshuffling algorithm is (Mohtashami et al., 2022; Lu et al., 2022b; Mishchenko et al., 2020) $\mathcal{O}\left(LF_0 n/T + \left(LF_0\sqrt{n}\sigma_{\text{SGD}}/T\right)^{\frac{2}{3}}\right)$, which is n times worse than our rate in the fist term, however it is $\sqrt{n}$ better in the leading stochastic term. It remains an open question whether it is possible to remove n from the first term of convergence without negatively affecting the leading stochastic term.

5.3. Discussion

Metric of convergence. In Theorem 5.1 we guarantee the convergence of $\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2$. This is a standard metric for analysing convergence of most of the established optimization algorithms: GD, SGD, Momentum SGD, second-order and high-order methods, and many others.

We would like to highlight here that prior works on incremental gradient methods proved convergence for the different metric. For example, (Lu et al., 2022b) provided convergence guarantees for $\min_{t=0,\ldots T}\left\|\nabla f(\mathbf{x}_t)\right\|^2$, and (Mishchenko et al., 2020; Mohtashami et al., 2022) provided convergence guarantees for $\frac{n}{T}\sum_{k=0,\ldots\lfloor T/n\rfloor}\left\|\nabla f(\mathbf{x}_{nk})\right\|^2$.

Parameter τ. We would like to highlight that parameter τ is not a hyperparameter of the algorithm (2) that we study. The parameter τ is introduced only in the theoretical analysis. In Theorem 5.1 we have to set parameter τ to a specific value ($\tau = \left\lfloor\frac{1}{8\sqrt{3}L\gamma}\right\rfloor$) in order to obtain the improved convergence guarantees, but this does not restrict the algorithm in any way, nor it restricts any of its hyperparameters.

5.4. Future directions

Below we highlight the open questions left for future work.

Improving the rate of RR. As highlighted in Section 5.2, using our proposed analysis, the resulting convergence rate for random reshuffling algorithm is not strictly better than the prior convergence rate. It is an interesting question for future work to understand whether it is possible to provide an improved analysis for random reshuffling algorithm, using our proof technique.

Optimality of the rates. While we provide improved convergence results for IG and SS algorithm, it remains an open question whether our convergence guarantees are tight. Understanding lower bounds for incremental gradient algorithms for non-convex smooth functions is an interesting question for future work.

6. Experiments

In this section, we present illustrative numerical experiments comparing different strategies for selecting stochastic gradients: SGD (sampling gradients with replacement), Single Shuffle (SS, using one random permutation for all epochs), and Random Reshuffling (RR, generating permutation for each new epoch). We demonstrate that both of shuffle strategies are not only beneficial due to simpler and faster implementations, but also achieve comparable or even better convergence than plain SGD.

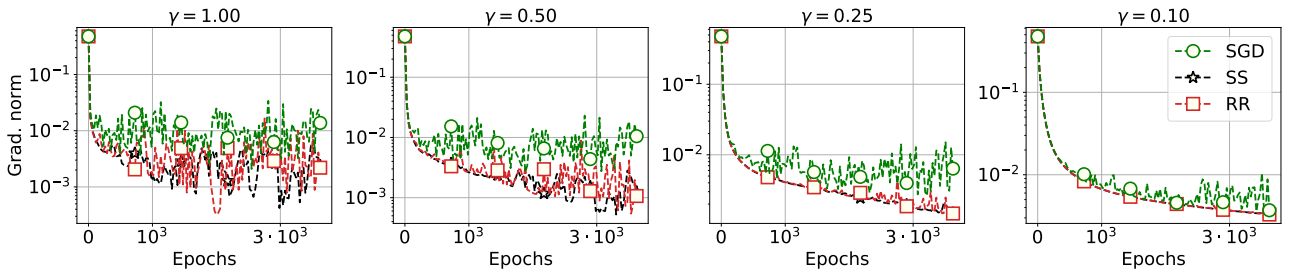


Figure 2. Convergence curves for logistic regression on the `Australian` dataset (Chang & Lin, 2011). Random Reshuffling (RR) and Single Shuffling (SS) are faster than SGD across varying learning rates.

Quadratic objectives. We first consider synthetic quadratic functions of the form $f_i(\mathbf{x}) = \frac{1}{2}\langle \mathbf{A}\mathbf{x}, \mathbf{x}\rangle - \langle \mathbf{b}, \mathbf{x}\rangle + \langle \mathbf{u}_i, \mathbf{x}\rangle$, for $1 \leq i \leq n$, where $\mathbf{A} = \mathbf{A}^\top \succeq 0$ is a given matrix and $\mathbf{b} \in \mathbb{R}^d$ is a fixed vector, while vectors $\{\mathbf{u}_i\}_{i=1}^n$ are generated randomly and rescaled to be zero-mean $\sum_{i=1}^n \mathbf{u}_i = \mathbf{0}$, and have the desired variance $\sigma_{\mathrm{SGD}}^2 = 0.01$.

In Figure 1(a) we depict convergence curves for various learning rates $\gamma \in \{5 \cdot 10^{-2}, 10^{-2}, 5 \cdot 10^{-1}\}$. In all the settings, both of the shuffle variants (SS and RR) outperform classical SGD, with the improvement being larger when the stepsize is smaller. This correspond to our theory in Equation (8), as the smaller γ, the larger the difference between $L\gamma \min\{1, L\gamma n\} n \sigma_{\mathrm{SGD}}^2$ (incremental gradient rate) and $L\gamma \sigma_{\mathrm{SGD}}^2$ (SGD rate).

In Figure 1(b) we additionally investigate the convergence properties of SS, RR and SGD across varying the number of components n in (1). We fix the target gradient norm to $\varepsilon = 10^{-2}$ and measure how many iterations $T_{\min}(n, \varepsilon)$ it takes for each of the methods to achieve this target accuracy. We tune the stepsize over the fixed grid separately for each method, and for each n. We repeat each experiments 30 times and plot the mean and 95% confidence intervals for $T_{\min}(n, \varepsilon)$.

We can see that both RR and SS *outperform* the classical SGD and the improvement is the most significant for smaller values of n.

Logistic regression. In Figure 2, we consider the problem of training the logistic regression model on machine learning datasets from (Chang & Lin, 2011)[1]. The objective components are convex and of the following form $f_i(\mathbf{x}) = \log(1 + e^{-y_i \langle \mathbf{a}_i, \mathbf{x}\rangle})$ where $\{\mathbf{a}_i\}_{i=1}^n$ are the feature vectors from $\mathbb{R}^d$ and $\{y_i\}_{i=1}^n$ are the labels from $\{\pm 1\}$. We compare RR, SS and SGD across four different learning rates $\gamma \in \{1, 0.5, 0.25, 0.1\}$. The results are shown in Fig. 2 and in the Appendix. We see that the Random Shuffle (RR) and Single Shuffle (SS) methods show significantly better performance than SGD, in all considered cases. Therefore, in practice, it seems indeed reasonable to always use the shuffle strategies instead of the classical sampling with replacement.

Neural network training. We train a three-layer neural network (one convolutional layer and two fully-connected layers with $\tanh$ activation functions) with the total number of parameters $d = 140697$. For our data we use a random subset of MNIST dataset of size 1000. After each epoch, we evaluate the norm of the full gradient for the entire model. The result are shown in Fig. 3. We see that the Random Reshuffling (RR) works at least as well as SGD or even better. The smaller the learning rate γ, the larger the improvement as predicted by our theory. We see that the convergence of RR is also more stable.[2]

Finally, we show experimental results on CIFAR dataset with `resnet18` architecture (He et al., 2016) (the total number of parameters is $d = 11181642$). In our training, we sample batches of a fixed size 256, comparing different strategies of sampling the data (SGD with replacements, Single Shuffle, and Random Reshuffling). The results are shown in Figure 4. We see that both shuffling strategies are better than the plain SGD. The best performance is achieved by Single Shuffle (SS).

6.1. Discussion

We observed that in most of cases SS and RR have similar practical performance (see Fig. 2, 3, 6, 7). Sometimes SS surpasses RR (Fig. 4), and sometimes RR surpasses SS (Fig. 1(a)). Regarding the practical choice between the two, in our opinion, the SS algorithm remains more preferable due to the effectiveness of a fixed data order: it eliminates the need to reshuffle the data after each epoch, which means the data can be physically distributed across different devices.

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

[2] In the Appendix, we provide full details on our experimental setup, computational environment, network architectures, as well as additional experiments.

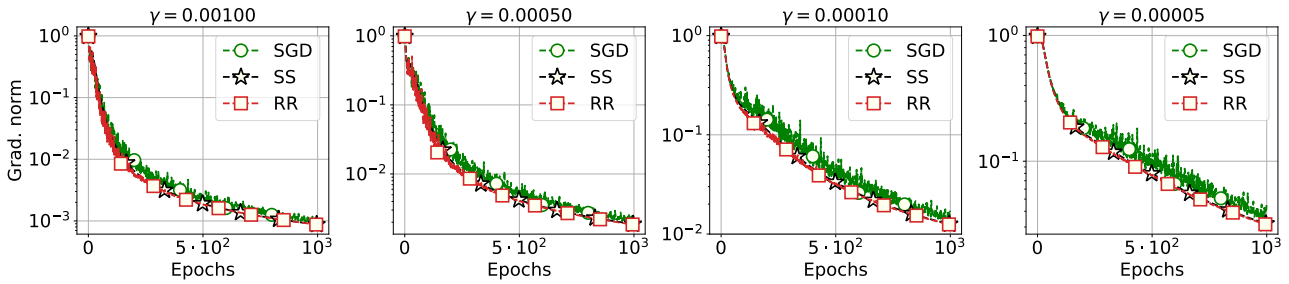


Figure 3. Training the neural network model on a subset of MNIST dataset of size 1000. Random Reshuffling (RR) and Single Shuffling (SS) are better than SGD.

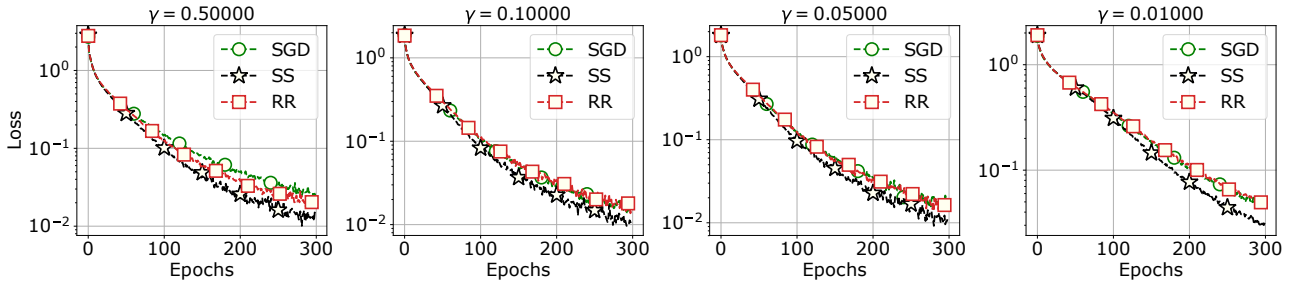


Figure 4. Training the neural network model on CIFAR dataset. Single Shuffle (SS) shows the best performance.

7. Conclusion

We present a framework for analyzing SGD algorithms under arbitrary data orderings. For incremental gradient descent and single shuffle algorithms we improve the previously best known convergence rates. This improvement is in part because our framework allows to chose larger stepsizes than in previous analyses because we can consider shorter correlation periods. Our study highlights the benefits of using SGD with single shuffling and provides new insights into its convergence properties for non-convex smooth optimization.

Acknowledgments

AK would like to thank Zachary Charles for the useful discussions that lead to the idea of this project. We also thank Amirkeivan Mohtashami for discussions, as well as Vinitra Swamy, Mary-Anne Hartley, and Alexander Hägele for their suggestions on manuscript writing. We are grateful to Cristobal Guzman, Grigory Malinovsky, and Peter Richtárik for their useful comments on the first version of our manuscript. AK is supported by a Google PhD Fellowship. ND is supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 22.00133.

Impact Statement

Our work is focused on improving the theoretical understanding of existing algorithms, we do not feel specific societal consequences must be specifically highlighted here.

References

Ahn, K., Yun, C., and Sra, S. Sgd with shuffling: optimal rates without component convexity and large epoch requirements, 06 2020.

Ajalloeian, A. and Stich, S. U. On the convergence of SGD with biased gradients. *arXiv preprint arXiv:2008.00051*, 2020.

Bengio, Y. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks*, 2012.

Bertsekas, D. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization*, 2010, 07 2015.

Bertsekas, D. P. Incremental proximal methods for large scale convex optimization. *Mathematical programming*, 129(2):163–195, 2011.

Bottou, L. Curiously fast convergence of some stochastic gradient descent algorithms. 2009.

Bubeck, S. *Convex Optimization: Algorithms and Complexity*. 01 2015. ISBN 9781601988614. doi: 10.1561/9781601988614.

Cha, J., Lee, J., and Yun, C. Tighter lower bounds for shuffling sgd: Random permutations and beyond. *arXiv preprint arXiv:2303.07160*, 2023.

Chang, C.-C. and Lin, C.-J. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

Cho, Y. J., Sharma, P., Joshi, G., Xu, Z., Kale, S., and Zhang, T. On the convergence of federated averaging with cyclic client participation. *arXiv preprint arXiv:2302.03109*, 2023.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N. M., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B. C., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., García, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Díaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K. S., Eck, D., Dean, J., Petrov, S., and Fiedel, N. Palm: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311, 2022.

Dekel, O., Gilad-Bachrach, R., Shamir, O., and Xiao, L. Optimal distributed online prediction using mini-batches. *J. Mach. Learn. Res.*, 13(null):165–202, jan 2012. ISSN 1532-4435.

Duchi, J., Agarwal, A., Johansson, M., and Jordan, M. Ergodic mirror descent. *SIAM Journal on Optimization*, 22, 05 2011. doi: 10.1109/Allerton.2011.6120236.

Eichner, H., Koren, T., McMahan, B., Srebro, N., and Talwar, K. Semi-cyclic stochastic gradient descent. In *International Conference on Machine Learning*, pp. 1764–1773. PMLR, 2019.

Even, M. Stochastic gradient descent under markovian sampling schemes, 2023.

Feder Cooper, A., Guo, W., Pham, K., Yuan, T., Ruan, C. F., Lu, Y., and De Sa, C. Cd-grab: Coordinating distributed example orders for provably accelerated training. *arXiv e-prints*, pp. arXiv–2302, 2023.

Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. 06 2017.

Grippo, L. A class of unconstrained minimization methods for neural network training. *Optimization Methods and Software*, 4(2):135–150, 1994.

Gürbüzbalaban, M., Ozdaglar, A., and Parrilo, P. Why random reshuffling beats stochastic gradient descent. *Mathematical Programming*, 186, 10 2015. doi: 10.1007/s10107-019-01440-w.

Gürbüzbalaban, M., Ozdaglar, A., and Parrilo, P. A. Convergence rate of incremental gradient and incremental newton methods. *SIAM Journal on Optimization*, 29(4):2542–2565, 2019.

Haochen, J. and Sra, S. Random shuffling beats SGD after finite epochs. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2624–2633. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/haochen19a.html.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Jain, P., Nagaraj, D. M., and Netrapalli, P. Sgd without replacement: Sharper rates for general smooth convex functions. In *International Conference on Machine Learning*, 2019.

Johansson, B., Rabi, M., and Johansson, M. A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Optimization*, 20:1157–1170, 01 2009. doi: 10.1137/08073038X.

Kohonen, T. An adaptive associative memory principle. *IEEE Transactions on Computers*, C-23(4):444–445, 1974. doi: 10.1109/T-C.1974.223960.

Koloskova, A., Loizou, N., Boreiri, S., Jaggi, M., and Stich, S. A unified theory of decentralized SGD with changing topology and local updates. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5381–5393. PMLR, 13–18 Jul 2020.

URL https://proceedings.mlr.press/v119/koloskova20a.html.

Koloskova, A., McKenna, R., Charles, Z., Rush, K., and McMahan, B. Convergence of gradient descent with linearly correlated noise and applications to differentially private learning, 2023. URL https://arxiv.org/abs/2302.01463.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lan, G. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133:1–33, 06 2012. doi: 10.1007/s10107-010-0434-y.

Lan, G. *First-order and stochastic optimization methods for machine learning*, volume 1. Springer, 2020.

LeCun, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Lu, Y., Guo, W., and Sa, C. D. Grab: Finding provably better data permutations than random reshuffling. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022a. URL https://openreview.net/forum?id=nDemfqKHTpK.

Lu, Y., Meng, S. Y., and Sa, C. D. A general analysis of example-selection for stochastic gradient descent. In *International Conference on Learning Representations*, 2022b. URL https://openreview.net/forum?id=7gWSJrP3opB.

Luo, Z.-Q. On the convergence of the lms algorithm with adaptive learning rate for linear feedforward networks. *Neural Computation*, 3(2):226–245, 1991.

Malinovsky, G., Mishchenko, K., and Richtárik, P. Server-side stepsizes and sampling without replacement provably help in federated optimization. *arXiv preprint arXiv:2201.11066*, 2022.

Mania, H., Pan, X., Papailiopoulos, D., Recht, B., Ramchandran, K., and Jordan, M. I. Perturbed iterate analysis for asynchronous stochastic optimization. *SIAM Journal on Optimization*, 27(4):2202–2229, 2017. doi: 10.1137/16M1057000. URL https://doi.org/10.1137/16M1057000.

Mishchenko, K., Khaled, A., and Richtárik, P. Random reshuffling: Simple analysis with vast improvements. *Advances in Neural Information Processing Systems*, abs/2006.05988, 2020.

Mishchenko, K., Khaled, A., and Richtárik, P. Proximal and federated random reshuffling. In *International Conference on Machine Learning*, pp. 15718–15749. PMLR, 2022.

Mohtashami, A., Stich, S. U., and Jaggi, M. Characterizing & finding good data orderings for fast convergence of sequential gradient methods. *ArXiv*, abs/2202.01838, 2022.

Nedić, A. and Bertsekas, D. Convergence rate of incremental subgradient algorithms. *Stochastic optimization: algorithms and applications*, pp. 223–264, 2001.

Nedic, A. and Bertsekas, D. P. Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12(1):109–138, 2001.

Nesterov, Y. *Lectures on convex optimization*, volume 137. Springer, 2018.

Nguyen, L., Tran-Dinh, Q., Phan, D., Nguyen, P., and van Dijk, M. A unified convergence analysis for shuffling-type gradient methods, 02 2020.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Rajput, S., Gupta, A., and Papailiopoulos, D. Closing the convergence gap of SGD without replacement. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7964–7973. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/rajput20a.html.

Recht, B. and Ré, C. Toward a noncommutative arithmetic-geometric mean inequality: Conjectures, case-studies, and consequences. In *Conference on Learning Theory*, pp. 11–1. JMLR Workshop and Conference Proceedings, 2012.

Recht, B. and Ré, C. Parallel stochastic gradient algorithms for large-scale matrix completion. *Mathematical Programming Computation*, 5, 04 2011. doi: 10.1007/s12532-013-0053-8.

Robbins, H. and Monro, S. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951. doi: 10.1214/aoms/1177729586. URL https://doi.org/10.1214/aoms/1177729586.

Rodomanov, A. and Kropotov, D. A superlinearly-convergent proximal Newton-type method for the optimization of finite sums. In *International Conference on Machine Learning*, pp. 2597–2605, 2016.

Sadiev, A., Malinovsky, G., Gorbunov, E., Sokolov, I., Khaled, A., Burlachenko, K., and Richtárik, P. Federated optimization algorithms with random reshuffling and gradient compression. *arXiv preprint arXiv:2206.07021*, 2022.

Safran, I. and Shamir, O. How good is sgd with random shuffling? In Abernethy, J. and Agarwal, S. (eds.), *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pp. 3250–3284. PMLR, 09–12 Jul 2020. URL https://proceedings.mlr.press/v125/safran20a.html.

Safran, I. and Shamir, O. Random shuffling beats SGD only after many epochs on ill-conditioned problems. *CoRR*, abs/2106.06880, 2021. URL https://arxiv.org/abs/2106.06880.

Shalev-Shwartz, S., Singer, Y., Srebro, N., and Cotter, A. Pegasos: primal estimated sub-gradient solver for SVM. *Mathematical Programming*, 127(1):3–30, 2011.

Stich, S. U. and Karimireddy, S. P. The error-feedback framework: Better rates for SGD with delayed gradients and compressed updates. *J. Mach. Learn. Res.*, 21(1), jun 2022. ISSN 1532-4435.

Sun, R.-Y. Optimization for deep learning: An overview. *Journal of the Operations Research Society of China*, 8 (2):249–294, 2020.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023.

Widrow, B. and Hoff, M. Adaptive switching circuitsinstitute of radio engineers, western electronic show and convention record, part 4.(pp. 96–104). new york, ny: Institute of radio engineers, 1960.

Yun, C., Rajput, S., and Sra, S. Minibatch vs local sgd with shuffling: Tight convergence bounds and beyond. *arXiv preprint arXiv:2110.10342*, 2021a.

Yun, C., Sra, S., and Jadbabaie, A. Can single-shuffle sgd be better than reshuffling sgd and gd? *ArXiv*, abs/2103.07079, 2021b.

Yun, C., Rajput, S., and Sra, S. Minibatch vs local SGD with shuffling: Tight convergence bounds and beyond. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=LdlwbBP2mlq.

A. Additional Experiments

A.1. Variance Estimation

First, we present an empirical study of different variance parameters (see Section 4.3), that are used to bound the convergence rates of stochastic gradient descent methods. For that, we take the `w1a` dataset from LibSVM (Chang & Lin, 2011) and consider the set of gradients $\{\nabla f_i(\mathbf{x}_0)\}_{i=1}^n$ for the Logistic Regression model (10), evaluated at a fixed point $\mathbf{x}_0 = \mathbf{0}$. We have an estimate for the standard variance used in the classic SGD analysis:

$$\sigma^2_{\text{SGD}} \quad := \quad \frac{1}{n}\sum_{j=1}^{n} \|\nabla f_j(\mathbf{x}_0) - \nabla f(\mathbf{x}_0)\|^2.$$

Then, for a randomly (uniformly) selected data ordering $(i_1, \ldots, i_n)$, we compute the quantities:

$$\xi_\tau\Big[(i_1, \ldots, i_n)\Big] \quad := \quad \Big\|\sum_{t=0}^{\tau} (\nabla f_{i_t}(\mathbf{x}_0) - \nabla f(\mathbf{x}_0))\Big\|^2, \qquad 1 \le \tau \le n,$$

and use them for estimating our variance parameter, as follows:

$$\sigma^2_\tau \quad := \quad \max_{j=0,\ldots,\tau-1} \bar{\xi}_j,$$

where $\bar{\xi}_j$ is the empirical mean estimated with 100 samples of data orderings. The result of our computations is shown in Figure 5. We see that the value of σ^2_τ, that is employed in our analysis of the Shuffle SGD methods is significantly better than its counterpart $n\sigma^2_{\text{SGD}}$ from the classic SGD with replacement. Hence, these observations empirically confirm superiority of the Shuffle strategies in practice.

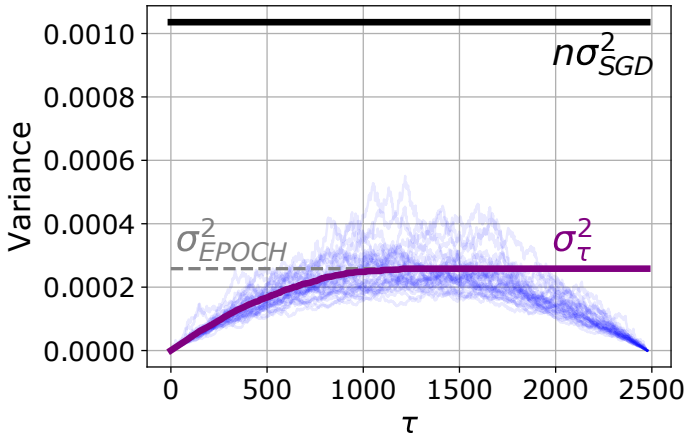


Figure 5. Variance estimation for the Logistic Regression model on `w1a` dataset. Our variance parameter σ^2_τ is significantly better than its corresponding upper bound $n\sigma^2_{\text{SGD}}$ used in classical SGD as well as in some prior works of analysing Shuffle SGD strategies (Mishchenko et al., 2020). When τ is smaller than $\frac{n}{2}$ our variance parameter σ^2_τ is also better than σ^2_{EPOCH} used to analyse Shuffle SGD in (Mohtashami et al., 2022; Lu et al., 2022b).

A.2. Logistic Regression

In this section we present experimental comparison of SGD with Single Shuffling and Random Reshuffling for training Logistic Regression model on several other datasets from LibSVM (Chang & Lin, 2011). Our objective has the following form,

$$\min_{\mathbf{x}\in\mathbb{R}^d}\Big[f(\mathbf{x}) \quad := \quad \frac{1}{n}\sum_{i=1}^{n} \log(1 + e^{-y_i\langle \mathbf{a}_i, \mathbf{x}\rangle})\Big], \tag{10}$$

where $\{\mathbf{a}_i\}_{i=1}^n$ are the vectors of features from $\mathbb{R}^d$ and $\{y_i\}_{i=1}^n$ are the training labels from $\{\pm 1\}$. Thus, our objective is smooth and convex, but not strongly convex. We apply all the methods starting from $\mathbf{x}_0 = \mathbf{0}$ and using a constant

stepsize $\gamma > 0$. We vary several values for γ, that are better suitable for each particular dataset. We show how the full gradient norm $\|\nabla f(\mathbf{x}_t)\|$ changes with iterations $t \geq 0$. One Epoch is equal to n iterations. The datasets we use are: `ionosphere` ($d = 34, n = 351$), `breast-cancer` ($d = 10, n = 683$), `a9a` ($d = 123, n = 32561$), randomly shrinked `w1a` ($d = 300, n = 500$), `rcv1` ($d = 47236, n = 20242$). The methods are implemented in Python 3.

In Figure 7, we see that the methods with Random Reshuffling (RR) and Single Shuffling (SS) constantly demonstrate better convergence behaviour for this problem, as compared to the classical SGD that uses sampling with replacement. For a fixed value γ, RR and SS strategies provide the method with a smaller variance of stochastic gradients, which in turn results in a more stable and faster convergence. This confirms our theory.

A.3. Neural Networks

In this section, we present our experimental results on training Neural Network models for MNIST (LeCun, 1998) and CIFAR (Krizhevsky et al., 2009) datasets. For MNIST, we used a small simple architecture consisting of one convolutional layer (with 3 output channels and the kernel of size 2) and two consequent fully connected layers with 64 and 10 output neurons respectively. Therefore, in total our model has $d = 140697$ parameters to train. We used smooth `tanh` activation function between layers. After each Epoch, we compute the full gradient norm evaluated over the entire training data. The results for a random selection of MNIST dataset of size $n = 1000$ are shown in Figure 3 and for $n = 500$ in Figure 6.

We see that all the methods posses similar convergence rates, while the convergence behaviour of RR and SS strategies is more stable than that of the SGD that uses sampling with replacement.

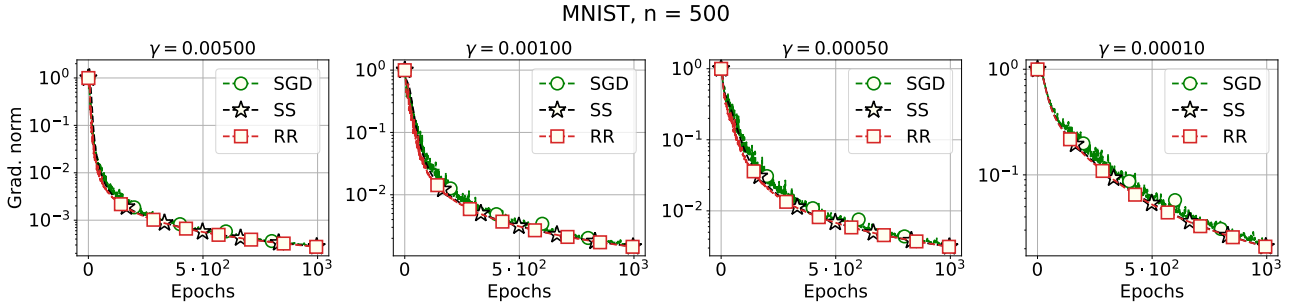


Figure 6. Training the neural network model on MNIST dataset. Random Reshuffling (RR) works always the same or better than SGD.

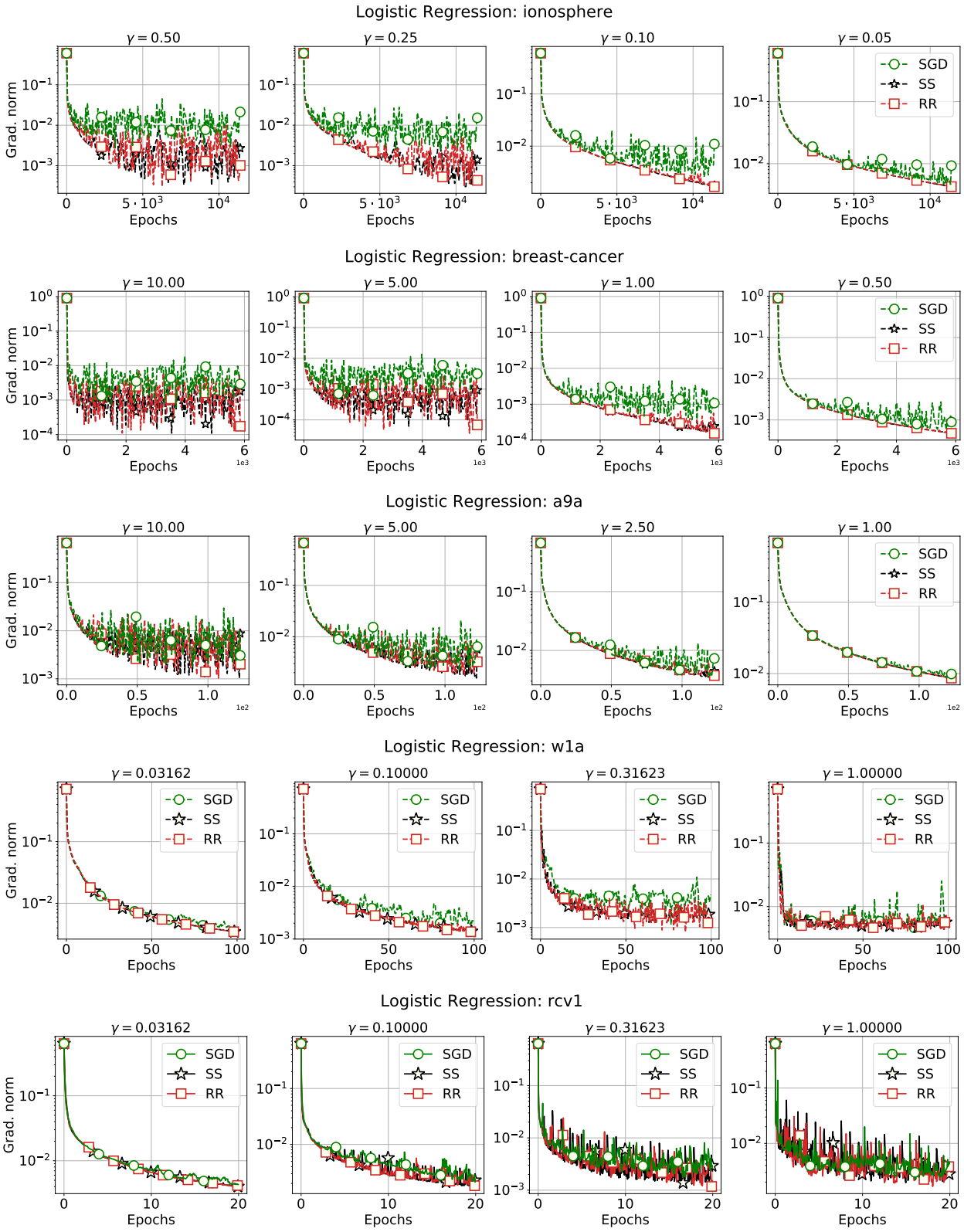


Figure 7. Convergence curves for logistic regression on the real data. Random Reshuffling (RR) and Single Shuffling (SS) work always better than SGD across varying learning rates.

B. Proofs

We first restate L-smoothness condition from Assumption 4.1.

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\| \ , \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d \, . \tag{11}$$

B.1. Useful inequalities

Lemma B.1. *For any finite set of vectors* $\{\mathbf{a}_i\}_{i=1}^n \subset \mathbb{R}^d$,

$$\left\| \sum_{i=1}^n \mathbf{a}_i \right\|^2 \leq n \sum_{i=1}^n \|\mathbf{a}_i\|^2 \, . \tag{12}$$

Lemma B.2. *For any two vectors* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ *and for all* $\alpha > 0$,

$$2\langle \mathbf{a}, \mathbf{b} \rangle \leq \alpha \|\mathbf{a}\|^2 + \alpha^{-1} \|\mathbf{b}\|^2 \, . \tag{13}$$

B.2. Main Lemma

Our proof is based on a technique called *perturbed iterate analysis* that analyzes a sequence of virtual iterates (Mania et al., 2017; Stich & Karimireddy, 2022). Recently, (Koloskova et al., 2023) proposed a modified virtual sequence with restart iterations and conducted an analysis for gradients perturbed with constant noise patterns (independent of $\mathbf{x}_t$). Here, we extend their analysis for arbitrary noise perturbations (that can depend on the iterates $\mathbf{x}_t$).

For our analysis we use the restart virtual sequence $\{\tilde{\mathbf{x}}_t\}_{t \geq 0}$, starting from $\tilde{\mathbf{x}}_0 = \mathbf{x}_0$ and defined as follows:

$$\begin{aligned} \tilde{\mathbf{x}}_{t+1} &= \tilde{\mathbf{x}}_t - \gamma \nabla f(\mathbf{x}_t) && \text{if } (t+1) \bmod \tau \neq 0 \\ \tilde{\mathbf{x}}_{t+1} &= \mathbf{x}_{t+1} && \text{if } (t+1) \bmod \tau = 0 \end{aligned} \tag{14}$$

where $\tau = \Theta\big(\frac{1}{L\gamma}\big)$ is our key parameter. Note that in (14) we use the *full gradients* $\nabla f(\cdot)$ evaluated at the iterates of our Algorithm 2. We denote by $r(t)$ the closest restart iteration to t, i.e.

$$r(t) \quad = \quad \lfloor \tfrac{t}{\tau} \rfloor \tau \quad = \quad t - t \bmod \tau.$$

For simplifying the presentation, we also denote, for $0 \leq t \leq T$:

$$\phi_t(\mathbf{x}) \quad = \quad \left\| \sum_{j=r(t)}^{t} \big(\nabla f(\mathbf{x}) - \nabla f_{i_j}(\mathbf{x}) \big) \right\|^2 \tag{15}$$

and

$$\bar{\phi}_t(\mathbf{x}) \quad = \quad \begin{cases} 0, & \text{if } t = r(t), \\ \left\| \sum_{j=r(t)}^{t-1} \big(\nabla f(\mathbf{x}) - \nabla f_{i_j}(\mathbf{x}) \big) \right\|^2, & \text{otherwise.} \end{cases} \tag{16}$$

Since, for $r(t) \leq i \leq r$, we have $r(i) = r(t)$, we conclude that

$$\phi_i(\mathbf{x}) \quad = \quad \left\| \sum_{j=r(t)}^{i} \big(\nabla f(\mathbf{x}) - \nabla f_{i_j}(\mathbf{x}) \big) \right\|^2, \qquad r(t) \leq i \leq t,$$

and, in particular, $\bar{\phi}_t(\mathbf{x}) = \phi_{t-1}(\mathbf{x})$ wherever $t \neq r(t)$.

Note that due to Definition 4.2, we have, for all $0 \leq t \leq T$:

$$\mathbb{E}\,\phi_t(\mathbf{x}) \quad \leq \quad \sigma_{\mathrm{k},\tau}^2, \qquad \text{with} \qquad k = \tfrac{r(t)}{\tau} = \lfloor \tfrac{t}{\tau} \rfloor. \tag{17}$$

First, we prove a lemma to bound the distance between the virtual sequence $\tilde{\mathbf{x}}_{t+1}$ and the real sequence $\mathbf{x}_{t+1}$ from Algorithm (2). Since the virtual and real iterate sequences have the following updates:

$$\mathbf{x}_{t+1} = \mathbf{x}_{r(t)} - \gamma \sum_{j=r(t)}^{t} \nabla f_{i_j}(\mathbf{x}_j), \qquad\qquad \tilde{\mathbf{x}}_{t+1} = \mathbf{x}_{r(t)} - \gamma \sum_{j=r(t)}^{t} \nabla f(\mathbf{x}_j), \tag{18}$$

it holds:

$$\|\tilde{\mathbf{x}}_{t+1} - \mathbf{x}_{t+1}\|^2 \quad = \quad \gamma^2 \left\| \sum_{j=r(t)}^{t} \left(\nabla f(\mathbf{x}_j) - \nabla f_{i_j}(\mathbf{x}_j)\right) \right\|^2.$$

Hence, it is enough just to bound the expression in the right hand side.

Lemma B.3. *Under the same assumptions as in Theorem 5.1, we have, for any $0 \leq t \leq T$:*

$$\begin{aligned} \left\| \sum_{j=r(t)}^{t} \left(\nabla f(\mathbf{x}_j) - \nabla f_{i_j}(\mathbf{x}_j)\right) \right\|^2 \quad \leq \quad & 3\phi_t(\mathbf{x}_{r(t)}) + 48\gamma^2 L^2 \tau \sum_{j=r(t)}^{t} \phi_j(\mathbf{x}_{r(t)}) \\ & + 16\gamma^2\tau^3 L^2 \sum_{j=r(t)}^{t} \|\nabla f(\mathbf{x}_j)\|^2. \end{aligned} \tag{19}$$

Therefore, substituting that $\gamma \leq \frac{1}{8\sqrt{3}L\tau}$, we have

$$\begin{aligned} \left\| \sum_{j=r(t)}^{t} \left(\nabla f(\mathbf{x}_j) - \nabla f_{i_j}(\mathbf{x}_j)\right) \right\|^2 \quad \leq \quad & 3\phi_t(\mathbf{x}_{r(t)}) + \frac{1}{4\tau} \sum_{j=r(t)}^{t} \phi_j(\mathbf{x}_{r(t)}) \\ & + \frac{\tau}{12} \sum_{j=r(t)}^{t} \|\nabla f(\mathbf{x}_j)\|^2. \end{aligned} \tag{20}$$

Proof. Indeed, we have

$$\begin{aligned} & \left\| \sum_{j=r(t)}^{t} \left(\nabla f(\mathbf{x}_j) - \nabla f_{i_j}(\mathbf{x}_j)\right) \right\|^2 \\ & \overset{(12)}{\leq} \quad 3\left\| \sum_{j=r(t)}^{t} \left(\nabla f(\mathbf{x}_{r(t)}) - \nabla f_{i_j}(\mathbf{x}_{r(t)})\right) \right\|^2 + 3\left\| \sum_{j=r(t)}^{t} \left(\nabla f(\mathbf{x}_{r(t)}) - \nabla f(\mathbf{x}_j)\right) \right\|^2 \\ & \qquad + \; 3\left\| \sum_{j=r(t)}^{t} \left(\nabla f_{i_j}(\mathbf{x}_j) - \nabla f_{i_j}(\mathbf{x}_{r(t)})\right) \right\|^2 \\ & \overset{(15),(12),(11)}{\leq} \quad 3\phi_t(\mathbf{x}_{r(t)}) + 6\tau L^2 \sum_{j=r(t)}^{t} \left\|\mathbf{x}_{r(t)} - \mathbf{x}_j\right\|^2. \end{aligned}$$

We further need to estimate the term

$$\sum_{j=r(t)}^{t} \left\|\mathbf{x}_{r(t)} - \mathbf{x}_j\right\|^2.$$

Looking individually at each element of the sum and using the update rule (2), we obtain

$$\begin{aligned}
&\left\|\mathbf{x}_j - \mathbf{x}_{r(t)}\right\|^2 \\
&= \gamma^2 \left\|\sum_{l=r(t)}^{j} \nabla f_{i_l}(\mathbf{x}_l)\right\|^2 \overset{(12)}{\leq} 2\gamma^2 \left\|\sum_{l=r(t)}^{j} \left(\nabla f_{i_l}(\mathbf{x}_l) - \nabla f(\mathbf{x}_l)\right)\right\|^2 + 2\gamma^2 \left\|\sum_{l=r(t)}^{j} \nabla f(\mathbf{x}_l)\right\|^2 \\
&\overset{(12)}{\leq} 6\gamma^2 \left\|\sum_{l=r(t)}^{j} \left(\nabla f_{i_l}(\mathbf{x}_{r(t)}) - \nabla f(\mathbf{x}_{r(t)})\right)\right\|^2 + 6\gamma^2 \left\|\sum_{l=r(t)}^{j} \left(\nabla f_{i_l}(\mathbf{x}_{r(t)}) - \nabla f_{i_l}(\mathbf{x}_l)\right)\right\|^2 \\
&\quad + 6\gamma^2 \left\|\sum_{l=r(t)}^{j} \left(\nabla f(\mathbf{x}_{r(t)}) - \nabla f(\mathbf{x}_l)\right)\right\|^2 + 2\gamma^2 \left\|\sum_{l=r(t)}^{j} \nabla f(\mathbf{x}_l)\right\|^2 \\
&\overset{(15),(12),(11)}{\leq} 6\gamma^2 \phi_j(\mathbf{x}_{r(t)}) + 12\gamma^2\tau L^2 \sum_{l=r(t)}^{j} \left\|\mathbf{x}_{r(t)} - \mathbf{x}_l\right\|^2 + 2\gamma^2\tau \sum_{l=r(t)}^{j} \left\|\nabla f(\mathbf{x}_l)\right\|^2 \\
&\overset{j\leq t}{\leq} 6\gamma^2 \phi_j(\mathbf{x}_{r(t)}) + 12\gamma^2\tau L^2 \sum_{l=r(t)}^{t} \left\|\mathbf{x}_{r(t)} - \mathbf{x}_l\right\|^2 + 2\gamma^2\tau \sum_{l=r(t)}^{t} \left\|\nabla f(\mathbf{x}_l)\right\|^2 .
\end{aligned}$$

Thus,

$$\begin{aligned}
&\sum_{j=r(t)}^{t} \left\|\mathbf{x}_{r(t)} - \mathbf{x}_j\right\|^2 \\
&\leq 6\gamma^2 \sum_{j=r(t)}^{t} \phi_j(\mathbf{x}_{r(t)}) + 12\gamma^2\tau^2 L^2 \sum_{j=r(t)}^{t} \left\|\mathbf{x}_{r(t)} - \mathbf{x}_j\right\|^2 + 2\gamma^2\tau^2 \sum_{j=r(t)}^{t} \left\|\nabla f(\mathbf{x}_j)\right\|^2 .
\end{aligned}$$

Using that $\tau \leq \frac{1}{8\sqrt{3}L\gamma}$, we get that $12\gamma^2\tau^2L^2 \leq \frac{1}{16}$. Thus, the coefficient in front of the second term in the right hand side is smaller than $\frac{1}{16}$, and rearranging this term we obtain

$$\sum_{j=r(t)}^{t} \left\|\mathbf{x}_{r(t)} - \mathbf{x}_j\right\|^2 \leq \frac{32\gamma^2}{5} \sum_{j=r(t)}^{t} \phi_j(\mathbf{x}_{r(t)}) + \frac{32\gamma^2\tau^2}{15} \sum_{j=r(t)}^{t-1} \left\|\nabla f(\mathbf{x}_j)\right\|^2 .$$

We therefore conclude (using the trivial upper bounds $\frac{6\cdot 32}{5} \leq 48$ and $\frac{6\cdot 32}{15} \leq 16$) that

$$\begin{aligned}
&\left\|\sum_{j=r(t)}^{t} \left(\nabla f(\mathbf{x}_j) - \nabla f_{i_j}(\mathbf{x}_j)\right)\right\|^2 \\
&\leq 3\phi_t(\mathbf{x}_{r(t)}) + 48\gamma^2 L^2\tau \sum_{j=r(t)}^{t} \phi_j(\mathbf{x}_{r(t)}) + 16\gamma^2\tau^3 L^2 \sum_{j=r(t)}^{t} \left\|\nabla f(\mathbf{x}_j)\right\|^2 . \quad \square
\end{aligned}$$

B.3. Proof of Theorem 5.1

We analyze separately the iterations t for which the restarts do not happen: $(t+1) \bmod \tau \neq 0$, and the restart iterations: $(t+1) \bmod \tau = 0$.

Iterations without restarts. Using L-smoothness of f, that is implied by Assumption 4.1,

$$\begin{aligned} f(\tilde{\mathbf{x}}_{t+1}) &\leq f(\tilde{\mathbf{x}}_t) - \gamma\langle \nabla f(\tilde{\mathbf{x}}_t), \nabla f(\mathbf{x}_t)\rangle + \frac{L\gamma^2}{2}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \\ &= f(\tilde{\mathbf{x}}_t) - \frac{\gamma}{2}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2 - \frac{\gamma}{2}\left\|\nabla f(\mathbf{x}_t)\right\|^2 + \frac{\gamma}{2}\left\|\nabla f(\tilde{\mathbf{x}}_t) - \nabla f(\mathbf{x}_t)\right\|^2 \\ &\quad + \frac{L\gamma^2}{2}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \\ &\leq f(\tilde{\mathbf{x}}_t) - \frac{\gamma}{2}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2 - \frac{\gamma}{2}\left\|\nabla f(\mathbf{x}_t)\right\|^2 + \frac{\gamma L^2}{2}\left\|\tilde{\mathbf{x}}_t - \mathbf{x}_t\right\|^2 + \frac{L\gamma^2}{2}\left\|\nabla f(\mathbf{x}_t)\right\|^2. \end{aligned}$$

Employing bound (20) from Lemma B.3 for the distance between the real and virtual sequences[3]:

$$\left\|\tilde{\mathbf{x}}_t - \mathbf{x}_t\right\|^2 = \gamma^2 \left\| \sum_{j=r(t)}^{t-1} \left(\nabla f(\mathbf{x}_j) - \nabla f_{i_j}(\mathbf{x}_j)\right)\right\|^2, \tag{21}$$

we get (note that we have a shifted index $t \mapsto t-1$ in (20), and to cover formally the trivial case $t = r(t)$ we use $\bar{\phi}_t(\mathbf{x}_t)$ in the bound instead of $\phi_{t-1}(\mathbf{x}_t)$):

$$\begin{aligned} f(\tilde{\mathbf{x}}_{t+1}) \leq f(\tilde{\mathbf{x}}_t) - \frac{\gamma}{2}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2 - \frac{\gamma}{2}\left\|\nabla f(\mathbf{x}_t)\right\|^2 + \frac{3L^2\gamma^3}{2}\bar{\phi}_t(\mathbf{x}_{r(t)}) + \frac{L^2\gamma^3}{8\tau}\sum_{j=r(t)}^{t-1}\phi_j(\mathbf{x}_{r(t)}) \\ + \frac{\tau L^2\gamma^3}{24}\sum_{j=r(t)}^{t-1}\left\|\nabla f(\mathbf{x}_j)\right\|^2 + \frac{L\gamma^2}{2}\left\|\nabla f(\mathbf{x}_t)\right\|^2. \end{aligned} \tag{22}$$

Restart Iterations. Next, we analyse the iterations t for which restarts happens in virtual sequence, i.e. $(t+1) \mod \tau = 0$. First, we re-write the update as

$$\begin{aligned} \tilde{\mathbf{x}}_{t+1} \overset{(14)}{=} \mathbf{x}_{t+1} \overset{(2)}{=} \mathbf{x}_t - \gamma\nabla f_{i_t}(\mathbf{x}_t) = \tilde{\mathbf{x}}_t + (\mathbf{x}_t - \tilde{\mathbf{x}}_t) - \gamma\nabla f_{i_t}(\mathbf{x}_t) \\ = \tilde{\mathbf{x}}_t - \gamma\nabla f(\mathbf{x}_t) + \gamma\sum_{j=r(t)}^{t}\left(\nabla f(\mathbf{x}_j) - \nabla f_{i_j}(\mathbf{x}_j)\right), \end{aligned}$$

where we used that $\mathbf{x}_t - \tilde{\mathbf{x}}_t = \gamma\sum_{j=r(t)}^{t-1}\left(\nabla f(\mathbf{x}_j) - \nabla f_{i_j}(\mathbf{x}_j)\right)$ due to the definitions (see (18)).

We use L-smoothness of f that follows from Assumption 4.1, thus

$$\begin{aligned} f(\tilde{\mathbf{x}}_{t+1}) \leq f(\tilde{\mathbf{x}}_t) - \gamma\langle \nabla f(\tilde{\mathbf{x}}_t), \nabla f(\mathbf{x}_t) - \sum_{j=r(t)}^{t}\left(\nabla f(\mathbf{x}_j) - \nabla f_{i_j}(\mathbf{x}_j)\right)\rangle \\ + \frac{L}{2}\gamma^2\left\|\nabla f(\mathbf{x}_t) - \sum_{j=r(t)}^{t}\left(\nabla f(\mathbf{x}_j) - \nabla f_{i_j}(\mathbf{x}_j)\right)\right\|^2 \\ \overset{(12)}{\leq} f(\tilde{\mathbf{x}}_t) + \underbrace{\left(-\gamma\langle \nabla f(\tilde{\mathbf{x}}_t), \nabla f(\mathbf{x}_t)\rangle\right)}_{:=T_1} + \underbrace{\gamma\langle \nabla f(\tilde{\mathbf{x}}_t), \sum_{j=r(t)}^{t}\left(\nabla f(\mathbf{x}_j) - \nabla f_{i_j}(\mathbf{x}_j)\right)\rangle}_{:=T_2} \\ + L\gamma^2\left\|\nabla f(\mathbf{x}_t)\right\|^2 + \underbrace{L\gamma^2\left\|\sum_{j=r(t)}^{t}\left(\nabla f(\mathbf{x}_j) - \nabla f_{i_j}(\mathbf{x}_j)\right)\right\|^2}_{:=T_3}. \end{aligned}$$

[3]For convenience, we set everywhere that $\sum_{j=a}^{b}\ldots \equiv 0$, wherever $a > b$.

We further separately estimate terms T_1, T_2 and T_3. We have

$$\begin{aligned} T_1 &= \frac{\gamma}{2}\left\|\nabla f(\tilde{\mathbf{x}}_t)-\nabla f(\mathbf{x}_t)\right\|^2-\frac{\gamma}{2}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2-\frac{\gamma}{2}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \\ &\overset{(11)}{\leq} \frac{L^2\gamma}{2}\left\|\tilde{\mathbf{x}}_t-\mathbf{x}_t\right\|^2-\frac{\gamma}{2}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2-\frac{\gamma}{2}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \\ &\overset{(21)}{=} \frac{L^2\gamma^3}{2}\left\|\sum_{j=r(t)}^{t-1}\left(\nabla f(\mathbf{x}_j)-\nabla f_{i_j}(\mathbf{x}_j)\right)\right\|^2-\frac{\gamma}{2}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2-\frac{\gamma}{2}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \end{aligned}$$

Using bound (20) for the first term (with a shifted index $t \mapsto t-1$, thus using $\bar{\phi}_t(\mathbf{x}_t)$ to cover formally the trivial case $t = r(t)$), we obtain

$$\begin{aligned} T_1 \leq\ & \frac{3L^2\gamma^3}{2}\bar{\phi}_t(\mathbf{x}_{r(t)})+\frac{L^2\gamma^3}{8\tau}\sum_{j=r(t)}^{t-1}\phi_j(\mathbf{x}_{r(t)})+\frac{\tau L^2\gamma^3}{24}\sum_{j=r(t)}^{t-1}\left\|\nabla f(\mathbf{x}_j)\right\|^2 \\ & -\frac{\gamma}{2}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2-\frac{\gamma}{2}\left\|\nabla f(\mathbf{x}_t)\right\|^2. \end{aligned}$$

The second term T_2 can be bounded as follows:

$$\begin{aligned} T_2 &= \langle \nabla f(\tilde{\mathbf{x}}_t), \gamma\sum_{j=r(t)}^{t}\left(\nabla f(\mathbf{x}_j)-\nabla f_{\pi_j}(\mathbf{x}_j)\right)\rangle \\ &\overset{(13)\text{ with }\alpha:=32\sqrt{3}L}{\leq} \underbrace{\frac{1}{64\sqrt{3}L}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2}_{:=T_4}+\underbrace{16\sqrt{3}L\gamma^2\left\|\sum_{j=r(t)}^{t}\left(\nabla f(\mathbf{x}_j)-\nabla f_{\pi_j}(\mathbf{x}_j)\right)\right\|^2}_{:=T_5}. \end{aligned}$$

We estimate T_4 as [4]

$$\begin{aligned} 64\sqrt{3}T_4 = \frac{1}{L}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2 &= \frac{1}{L\tau}\sum_{j=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2 \\ &\overset{(12)}{\leq} \frac{2}{L\tau}\sum_{j=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_t)-\nabla f(\tilde{\mathbf{x}}_{t-j})\right\|^2+\frac{2}{L\tau}\sum_{j=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_{t-j})\right\|^2 \\ &\overset{(11)}{\leq} \frac{2}{L\tau}\sum_{j=0}^{\tau-1}L^2\left\|\tilde{\mathbf{x}}_t-\tilde{\mathbf{x}}_{t-j}\right\|^2+\frac{2}{L\tau}\sum_{j=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_{t-j})\right\|^2 \\ &\leq \frac{2\gamma^2L}{\tau}\sum_{j=0}^{\tau-1}\left\|\sum_{k=t-j}^{t-1}\nabla f(\mathbf{x}_k)\right\|^2+\frac{2}{L\tau}\sum_{j=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_{t-j})\right\|^2 \\ &\overset{(12)}{\leq} 2\gamma^2L\sum_{j=0}^{\tau-1}\sum_{k=t-j}^{t-1}\left\|\nabla f(\mathbf{x}_k)\right\|^2+\frac{2}{L\tau}\sum_{j=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_{t-j})\right\|^2 \\ &\leq 2\gamma^2L\tau\sum_{j=0}^{\tau-1}\left\|\nabla f(\mathbf{x}_{t-j})\right\|^2+\frac{2}{L\tau}\sum_{j=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_{t-j})\right\|^2. \end{aligned}$$

[4] On the second line we could use the equation $\|\mathbf{a}+\mathbf{b}\|^2 \leq (1+\alpha)\|\mathbf{a}\|^2+(1+\alpha^{-1})\|\mathbf{b}\|^2$ for some constant $\alpha < 1$ to get a better dependence on the constants.

Further using that $\tau = \left\lfloor \frac{1}{8\sqrt{3}L\gamma} \right\rfloor \geq 1$, which means both that $\tau \leq \frac{1}{8\sqrt{3}L\gamma}$ and[5] $\tau \geq \frac{1}{16\sqrt{3}L\gamma}$ we get

$$\frac{1}{L}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2 \leq \frac{\gamma}{4\sqrt{3}}\sum_{j=0}^{\tau-1}\left\|\nabla f(\mathbf{x}_{t-j})\right\|^2 + 32\sqrt{3}\gamma\sum_{j=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_{t-j})\right\|^2.$$

Thus,

$$T_4 = \frac{1}{64\sqrt{3}L}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2 \leq \frac{\gamma}{64\cdot 12}\sum_{j=0}^{\tau-1}\left\|\nabla f(\mathbf{x}_{t-j})\right\|^2 + \frac{\gamma}{2}\sum_{j=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_{t-j})\right\|^2.$$

It is left to estimate the terms $T_3 + T_5$, as follows:

$$\begin{aligned}
T_3 + T_5 \quad &\leq \quad 29L\gamma^2\left\|\sum_{j=r(t)}^{t}\left(\nabla f(\mathbf{x}_j) - \nabla f_{\pi_j}(\mathbf{x}_j)\right)\right\|^2 \\
&\overset{(20)}{\leq} \quad 87L\gamma^2\phi_t(\mathbf{x}_{r(t)}) + \frac{29L\gamma^2}{4\tau}\sum_{j=r(t)}^{t}\phi_j(\mathbf{x}_{r(t)}) + \frac{29\tau L\gamma^2}{12}\sum_{j=r(t)}^{t}\left\|\nabla f(\mathbf{x}_j)\right\|^2 \\
&\overset{\tau\leq\frac{1}{8\sqrt{3}L\gamma}}{\leq} \quad 87L\gamma^2\phi_t(\mathbf{x}_{r(t)}) + \frac{29L\gamma^2}{4\tau}\sum_{j=r(t)}^{t}\phi_j(\mathbf{x}_{r(t)}) + \frac{\gamma}{6}\sum_{j=r(t)}^{t}\left\|\nabla f(\mathbf{x}_j)\right\|^2.
\end{aligned}$$

Summing up estimations for T_1, T_2, T_3, T_4 and T_5. After summing up, the descent equation for the restart iterations is

$$\begin{aligned}
f(\tilde{\mathbf{x}}_{t+1}) \quad \leq \quad f(\tilde{\mathbf{x}}_t) \quad & \underbrace{\begin{array}{l} -\frac{\gamma}{2}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2 - \frac{\gamma}{2}\left\|\nabla f(\mathbf{x}_t)\right\|^2 + \frac{3L^2\gamma^3}{2}\bar{\phi}_t(\mathbf{x}_{r(t)}) \\ + \frac{L^2\gamma^3}{8\tau}\sum_{j=r(t)}^{t-1}\phi_j(\mathbf{x}_{r(t)}) + \frac{\tau L^2\gamma^3}{24}\sum_{j=r(t)}^{t-1}\left\|\nabla f(\mathbf{x}_j)\right\|^2 \end{array}}_{\text{bound for } T_1} \\
+ \quad L\gamma^2\left\|\nabla f(\mathbf{x}_t)\right\|^2 + & \underbrace{87L\gamma^2\phi_t(\mathbf{x}_{r(t)}) + \frac{29L\gamma^2}{4\tau}\sum_{j=r(t)}^{t}\phi_j(\mathbf{x}_{r(t)}) + \frac{\gamma}{6}\sum_{i=r(t)}^{t}\left\|\nabla f(\mathbf{x}_i)\right\|^2}_{\text{bound for } T_3+T_5} \\
+ \quad & \underbrace{\frac{\gamma}{64\cdot 12}\sum_{i=0}^{\tau-1}\left\|\nabla f(\mathbf{x}_{t-i})\right\|^2 + \frac{\gamma}{2}\sum_{i=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_{t-i})\right\|^2}_{\text{bound for } T_4}.
\end{aligned} \tag{23}$$

Summing up the descent equations (22) and (23) from all the iterations t. We will use that

$$\sum_{t=0}^{T}\frac{\tau L^2\gamma^3}{24}\sum_{i=r(t)}^{t-1}\left\|\nabla f(\mathbf{x}_i)\right\|^2 \quad \leq \quad \frac{\tau^2 L^2\gamma^3}{24}\sum_{t=0}^{T}\left\|\nabla f(\mathbf{x}_t)\right\|^2$$

[5] We note that for simplicity of presentation this lower bound is rough and could be refined if we want to improve dependence on the numerical constants.

Summing up (22) and (23) for $0 \le t \le T$, we get

$$\begin{aligned} f(\tilde{\mathbf{x}}_{T+1}) - f(\tilde{\mathbf{x}}_0) \le & -\frac{\gamma}{2}\sum_{t=0}^{T}\|\nabla f(\tilde{\mathbf{x}}_t)\|^2 - \Big(\frac{\gamma}{2} - L\gamma^2\Big)\sum_{t=0}^{T}\|\nabla f(\mathbf{x}_t)\|^2 + \frac{\tau^2\gamma^3L^2}{24}\sum_{t=0}^{T}\|\nabla f(\mathbf{x}_t)\|^2 \\ & + \frac{3L^2\gamma^3}{2}\sum_{t=0}^{T}\bar{\phi}_t(\mathbf{x}_{r(t)}) + \frac{L^2\gamma^3}{8\tau}\sum_{t=0}^{T}\sum_{j=r(t)}^{t-1}\phi_j(\mathbf{x}_{r(t)}) \\ & + \Big(\frac{\gamma}{6} + \frac{\gamma}{64\cdot 24}\Big)\sum_{t=0}^{T}\|\nabla f(\mathbf{x}_t)\|^2 + \frac{\gamma}{2}\sum_{t=0}^{T}\|\nabla f(\tilde{\mathbf{x}}_t)\|^2 \\ & + \sum_{k=1}^{\lceil\frac{T}{\tau}\rceil}\left[87L\gamma^2\phi_{k\tau-1}(\mathbf{x}_{(k-1)\tau}) + \frac{29L\gamma^2}{4\tau}\sum_{j=(k-1)\tau}^{k\tau-1}\phi_j(\mathbf{x}_{(k-1)\tau})\right], \end{aligned} \tag{24}$$

where on the first two lines are the terms that appears in both of the cases, and on the last two lines are the terms specific to the restart iteration cases, and thus happen once every τ iterations only.

It is left to: (i) sum up the coefficients in front of $\sum_{t=0}^{T}\|\nabla f(\mathbf{x}_t)\|^2$ and $\sum_{t=0}^{T}\|\nabla f(\tilde{\mathbf{x}}_t)\|^2$ (where the latter coeficient is equal to zero), (ii) take the expectation over the choosing orders i_t, and estimate terms with $\phi_t(\mathbf{x})$ through Definition 4.2.

First (i) we sup up the coefficients in front of $\sum_{t=0}^{T}\|\nabla f(\mathbf{x}_t)\|^2$, that is

$$\begin{aligned} -\tfrac{\gamma}{2} + L\gamma^2 + \tfrac{\tau^2\gamma^3L^2}{24} + \tfrac{\gamma}{6} + \tfrac{\gamma}{64\cdot 24} & \overset{\tau\le\frac{1}{8\sqrt{3}L\gamma}}{\le} -\tfrac{\gamma}{2} + L\gamma^2 + \tfrac{\gamma}{64\cdot 24\cdot 3} + \tfrac{\gamma}{6} + \tfrac{\gamma}{64\cdot 24} \\ & \overset{\gamma\le\frac{1}{8\sqrt{3}L}}{\le} -\tfrac{\gamma}{2} + \tfrac{\gamma}{8\sqrt{3}} + \tfrac{\gamma}{64\cdot 24\cdot 3} + \tfrac{\gamma}{6} + \tfrac{\gamma}{64\cdot 24} \\ & \le -\tfrac{\gamma}{4}. \end{aligned}$$

Second (ii), we note that for any $0 \le t \le T$ and $r(t) \le j \le t$, it holds that $\mathbb{E}\,\phi_j(\mathbf{x}_{r(t)}) \le \sigma_\tau^2$ for $k = \lfloor\frac{t}{\tau}\rfloor$ (see (17)). Thus, we can bound the second line of (24) as

$$\begin{aligned} \frac{3L^2\gamma^3}{2}\sum_{t=0}^{T}\mathbb{E}\,\bar{\phi}_t(\mathbf{x}_{r(t)}) + \frac{L^2\gamma^3}{8\tau}\sum_{t=0}^{T}\sum_{j=r(t)}^{t-1}\mathbb{E}\,\phi_j(\mathbf{x}_{r(t)}) &\le \frac{3L^2\gamma^3}{2}\sum_{t=0}^{T}\sigma_\tau^2 + \frac{L^2\gamma^3}{8}\sum_{t=0}^{T}\sigma_\tau^2 \\ &\le 2L^2\gamma^3\sum_{t=0}^{T}\sigma_\tau^2 \\ &\le 2L^2\gamma^3(T+1)\,\sigma_\tau^2\,. \end{aligned}$$

Similarly, for the last line of (24) we can estimate

$$\begin{aligned} & \sum_{k=1}^{\lceil\frac{T}{\tau}\rceil}\left[87L\gamma^2\,\mathbb{E}\,\phi_{k\tau-1}(\mathbf{x}_{(k-1)\tau}) + \frac{29L\gamma^2}{4\tau}\sum_{j=(k-1)\tau}^{k\tau-1}\mathbb{E}\,\phi_j(\mathbf{x}_{(k-1)\tau})\right] \\ \le \quad & 95L\gamma^2\sum_{k=0}^{\lfloor\frac{T}{\tau}\rfloor}\sigma_\tau^2 \quad \overset{\tau\ge\frac{1}{16\sqrt{3}L\gamma}}{\le} \quad 95\cdot 16\cdot\sqrt{3}\cdot L^2\gamma^3\tau\sum_{k=0}^{\lfloor\frac{T}{\tau}\rfloor}\sigma_\tau^2 \le 95\cdot 16\cdot\sqrt{3}\cdot L^2\gamma^3(T+1)\,\sigma_\tau^2\,. \end{aligned}$$

Putting back these calculations into (24), we get

$$f(\tilde{\mathbf{x}}_{T+1}) - f(\tilde{\mathbf{x}}_0) \le -\frac{\gamma}{4}\sum_{t=0}^{T}\|\nabla f(\mathbf{x}_t)\|^2 + AL^2\gamma^3(T+1)\,\sigma_\tau^2,$$

where $A = 2633$ is a numerical constant.

Rearranging, dividing by $T+1$, and using that $\tilde{\mathbf{x}}_0 = \mathbf{x}_0$, and that $f(\tilde{\mathbf{x}}_{T+1}) \geq f^\star$ we get

$$\frac{1}{T+1}\sum_{t=0}^{T}\|\nabla f(\mathbf{x}_t)\|^2 \leq \mathcal{O}\left(\frac{f(\mathbf{x}_0)-f^\star}{\gamma T} + L^2\gamma^2\frac{\tau}{T}\sigma_\tau^2\right).$$

C. Proofs of the bounds in Table 2

In this section we give upper bounds on σ_τ^2 for the special cases given in Table 2.

C.1. SGD, Example 3.1

Recall that according to Def. 4.2,

$$\sigma_\tau^2 = \sup_{\mathbf{x}\in\mathbb{R}^d}\max_{\substack{k=0,\ldots,\lfloor\frac{T}{\tau}\rfloor\\ j=0,\ldots,\tau-1}}\mathbb{E}\left[\left\|\sum_{t=k\tau}^{\min\{k\tau+j,T\}}\left(\nabla f_{i_t}(\mathbf{x})-\nabla f(\mathbf{x})\right)\right\|^2\middle|\, i_0,\ldots,i_{k\tau-1}\right].$$

Since in SGD each of i_t are sampled independently uniformly at random from $[n]$, the conditional expectation is equal to unconditional, and we also know that $\mathbb{E}\,\nabla f_{i_t}(\mathbf{x}) = \nabla f(\mathbf{x})$, and

$$\begin{aligned}
&\sup_{\mathbf{x}\in\mathbb{R}^d}\max_{\substack{k=0,\ldots,\lfloor\frac{T}{\tau}\rfloor\\ j=0,\ldots,\tau-1}}\mathbb{E}\left\|\sum_{t=k\tau}^{\min\{k\tau+j,T\}}\left(\nabla f_{i_t}(\mathbf{x})-\nabla f(\mathbf{x})\right)\right\|^2\\
&\quad= \sup_{\mathbf{x}_0\in\mathbb{R}^d}\max_{\substack{k=0,\ldots,\lfloor\frac{T}{\tau}\rfloor\\ j=0,\ldots,\tau-1}}\mathbb{E}\sum_{t=k\tau}^{\min\{k\tau+j,T\}}\|\nabla f_{i_t}(\mathbf{x})-\nabla f(\mathbf{x})\|^2\\
&\quad\overset{j=\tau}{\leq} \sup_{\mathbf{x}_0\in\mathbb{R}^d}\max_{k=0,\ldots,\lfloor\frac{T}{\tau}\rfloor}\mathbb{E}\sum_{t=k\tau}^{\min\{k\tau+\tau,T\}}\|\nabla f_{i_t}(\mathbf{x})-\nabla f(\mathbf{x})\|^2\\
&\quad\overset{(5)}{\leq} \max_{k=0,\ldots,\lfloor\frac{T}{\tau}\rfloor}\sum_{t=k\tau}^{\min\{k\tau+\tau,T\}}\sigma_{\mathrm{SGD}}^2\\
&\quad\leq \tau\,\sigma_{\mathrm{SGD}}^2\,.
\end{aligned}$$

This proves the first bound in Table 2 that $\sigma_\tau^2 \leq \tau\,\sigma_{\mathrm{SGD}}^2$ for the SGD algorithm.

C.2. Incremental Gradient and Single Shuffle, Example 3.2, Example 3.3

First, we note that if the interval between $k\tau$ and $k\tau+j$ contains the full epoch inside it, i.e. $\{k'n,\ldots,(k'+1)n\} \subseteq \{k\tau,\ldots,k\tau+j\}$ for an integer k', the gradients from this epoch cancel out with the full gradient. This is because $\sum_{i=1}^n \nabla f_{\pi_i}(\mathbf{x}) = \sum_{i=1}^n \nabla f_i(\mathbf{x}) = n\nabla f(\mathbf{x})$ and thus $\sum_{i=1}^n(\nabla f_{\pi_i}(\mathbf{x})-\nabla f(\mathbf{x})) = 0$.

Thus, w.l.o.g. we can assume that

$$\sum_{t=k\tau}^{k\tau+j}\left(\nabla f_{i_t}(\mathbf{x})-\nabla f(\mathbf{x})\right) = \sum_{i=j_1}^{n}\left(\nabla f_i(\mathbf{x})-\nabla f(\mathbf{x})\right) + \sum_{i=1}^{j_2}\left(\nabla f_i(\mathbf{x})-\nabla f(\mathbf{x})\right).$$

Moreover, if $j_1 < j_2$ then the full epoch will cancel with the full gradient $\nabla f(\mathbf{x})$ and this sum will be equal only to the intersecting part $\sum_{i=j_1}^{j_2}(\nabla f_i(\mathbf{x})-\nabla f(\mathbf{x}))$. Thus, we will just assume that

$$\sum_{t=k\tau}^{k\tau+j}\left(\nabla f_{i_t}(\mathbf{x})-\nabla f(\mathbf{x})\right) = \sum_{i\in\mathcal{S}}\left(\nabla f_i(\mathbf{x})-\nabla f(\mathbf{x})\right),$$

where $\mathcal{S} \subset [n]$, and thus $|\mathcal{S}| \leq \min\{j, n\} \leq \min\{\tau, n\}$. Therefore,

$$\begin{aligned}\left\|\sum_{t=k\tau}^{k\tau+j} (\nabla f_{i_t}(\mathbf{x}) - \nabla f(\mathbf{x}))\right\|^2 &= \left\|\sum_{i\in\mathcal{S}} (\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x}))\right\|^2 \\ &\overset{(12)}{\leq} |\mathcal{S}| \sum_{i\in\mathcal{S}} \|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \\ &\leq |\mathcal{S}| n \cdot \frac{1}{n}\sum_{i=1}^{n} \left\|\nabla f_i(\mathbf{x}_{r(t)}) - \nabla f(\mathbf{x}_{r(t)})\right\|^2 \\ &\leq |\mathcal{S}| n \cdot \sigma_{\mathrm{SGD}}^2 \\ &\leq \min\{\tau, n\} n \cdot \sigma_{\mathrm{SGD}}^2 \,.\end{aligned}$$

This proves the bound, $\sigma_{\mathrm{k},\tau}^2 \leq \min\{\tau, n\} n\, \sigma_{\mathrm{SGD}}^2$.

C.3. Random Shuffle, Example 3.4

In order to estimate σ_τ^2, we will first estimate $\mathbb{E}\left\|\sum_{t=k\tau}^{k\tau+j} (\nabla f_{i_t}(\mathbf{x}) - \nabla f(\mathbf{x}))\right\|^2$, for a fixed $0 \leq j \leq \tau - 1$, and then we will take the maximum over it. We need to consider only j such that $k\tau + j \leq T$.

First, we note that if the interval between $k\tau$ and $k\tau + j$ contains the full epoch inside it, i.e. $\{k'n, \ldots, (k'+1)n\} \subseteq \{k\tau, \ldots, k\tau + j\}$ for an integer k', the gradients from this epoch cancel out with the full gradient. This is because $\sum_{i=1}^n \nabla f_{\pi_i}(\mathbf{x}) = \sum_{i=1}^n \nabla f_i(\mathbf{x}) = n\nabla f(\mathbf{x})$ and thus $\sum_{i=1}^n (\nabla f_{\pi_i}(\mathbf{x}) - \nabla f(\mathbf{x})) = 0$.

Next, we note that the interval between $k\tau$ and $k\tau + j$ might overlap with more than one epochs (one epochs is equal to n iterations), but it can contain only at most two incomplete epochs. Thus, w.l.o.g., we consider that the interval $k\tau$ and $k\tau + j$ intersects with the two (not full) epochs, and the lengths of overlaps are equal to j_1 and j_2 correspondingly. And thus,

$$\sum_{t=k\tau}^{k\tau+j} (\nabla f_{i_t}(\mathbf{x}) - \nabla f(\mathbf{x})) = \sum_{t=1}^{j_1} \left(\nabla f_{\pi_t^1}(\mathbf{x}) - \nabla f(\mathbf{x})\right) + \sum_{t=1}^{j_2} \left(\nabla f_{\pi_t^2}(\mathbf{x}) - \nabla f(\mathbf{x})\right)$$

where π^1 and π^2 correspond to the two permutations, and $j_1 + j_2 \leq j + 1 \leq \tau$. Taking the norm and the expectation over permutations π^1 and π^2 (note that indices j_1 and j_2 are non-randomized fixed parameters that depend only on j, τ and n),

We further use that

$$\begin{aligned}&\left\|\sum_{t=1}^{j_1} \left(\nabla f_{\pi_t^1}(\mathbf{x}) - \nabla f(\mathbf{x})\right) + \sum_{t=1}^{j_2} \left(\nabla f_{\pi_t^2}(\mathbf{x}) - \nabla f(\mathbf{x})\right)\right\|^2 \\ &\leq 2\left\|\sum_{t=1}^{j_1} \left(\nabla f_{\pi_t^1}(\mathbf{x}) - \nabla f(\mathbf{x})\right)\right\|^2 + 2\left\|\sum_{t=1}^{j_2} \left(\nabla f_{\pi_t^2}(\mathbf{x}) - \nabla f(\mathbf{x})\right)\right\|^2\end{aligned}$$

Taking the conditional expectation, and estimating the two terms using similar calculations to the previous case of Incremental Gradient and Single Shuffle, we arrive that $\sigma_\tau^2 \leq 4\min\{\tau, n\} n \cdot \sigma_{\mathrm{SGD}}^2$.

C.4. Single function, Example 3.5

Since $i_t \equiv 1\ \forall t$, we have, using that $j \leq \tau$:

$$\left\|\sum_{t=k\tau}^{k\tau+j} (\nabla f_1(\mathbf{x}) - \nabla f(\mathbf{x}))\right\|^2 = \|j\,(\nabla f_1(\mathbf{x}) - \nabla f(\mathbf{x}))\|^2 = j^2\,\|\nabla f_1(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq \tau^2\,\sigma_{\mathrm{ONE}}^2\,.$$

D. Unavoidable bias in (9)

We now show that $\sigma^2_{\rm ONE}$ in (9) is an unavoidable bias. Assume that we have two functions $f_1(\mathbf{x}) = \frac{1}{2}\left\|\mathbf{x}-\mathbf{a}\right\|^2$ and $f_2(\mathbf{x}) = \frac{1}{2}\left\|\mathbf{x}+\mathbf{a}\right\|^2$ for some constant vector $\mathbf{a}$. Then,

$$\nabla f_1(\mathbf{x}) = \mathbf{x}-\mathbf{a} \qquad\qquad \nabla f(\mathbf{x}) = \mathbf{x}$$

and thus, $\sigma^2_{\rm ONE} = \left\|\mathbf{a}\right\|^2$. If we only optimize over the function f_1, then Algorithm (2) will converge to the optimum of f_1 which is $\mathbf{x}_1^\star = \mathbf{a}$. Then the norm of the full gradient $\left\|\nabla f(\mathbf{x}_1^\star)\right\|^2 = \left\|\mathbf{a}\right\|^2 = \sigma^2_{\rm ONE}$. Thus, the algorithm can converge only to the neighbourhood of the size $\sigma^2_{\rm ONE}$.

E. Stepsize tuning details

In this section we provide the stepsize tuning details for the results listed in Section 5.1. We start with the basic SGD algorithm, as our analysis recovers its standard rate of convergence (Ghadimi & Lan, 2013).

E.1. Basic SGD

Corollary E.1 (SGD, fixed number of iterations)**.** *Let each of the functions f_i be L-smooth (Assumption 4.1). Let i_t in Algorithm (2) be chosen as in Example 3.1, i.e. $i_t \sim [n]$ (uniformly and independently in every step). Let $T \geq 1$ be a fixed number of iterations. Then, after running Algorithm (2) for T iterations with the stepsize*

$$\gamma = \min\left\{\frac{1}{14L}, \left(\frac{F_0}{L\,\sigma^2_{\rm SGD}\,T}\right)^{1/2}\right\}, \tag{25}$$

the average gradient norm is bounded as follows:

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}\left(\frac{LF_0}{T} + \sqrt{\frac{LF_0\,\sigma^2_{\rm SGD}}{T}}\right),$$

where $F_0 = f(\mathbf{x}_0) - f^\star$, and $\mathcal{O}(\cdot)$ hides absolute numerical constants that do not depend on any of our parameters.

Proof. Using result of Theorem 5.1, and estimating $\sigma_\tau^2 \leq \tau\,\sigma^2_{\rm SGD}$ (see Table 2) we get

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}\left(\frac{F_0}{\gamma T} + L\gamma\,\sigma^2_{\rm SGD}\right), \tag{7}$$

valid for any $0 < \gamma \leq \frac{1}{14L} \leq \frac{1}{8\sqrt{3}L}$. We further can use Lemma 17 from (Koloskova et al., 2020) to tune the stepsize. For completeness of our presentation, we present the calculations below.

Note that in the right hand side of (7), we have a univariate convex function of γ, and our goal is to make it as small as possible. Therefore, we can minimize this right hand side with respect to constraints $0 < \gamma \leq \frac{1}{14L}$. Thus, taking the *optimal* stepsize $\gamma = \min\left\{\frac{1}{14L}, \left(\frac{F_0}{L\,\sigma^2_{\rm SGD}\,T}\right)^{1/2}\right\}$, we have to consider the two cases:

- $\gamma = \frac{1}{14L}$. This also means that $\gamma \leq \left(\frac{F_0}{L\,\sigma^2_{\rm SGD}\,T}\right)^{1/2}$. Substituting these estimation into (7) we get

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}\left(\frac{14LF_0}{T} + L\left(\frac{F_0}{L\,\sigma^2_{\rm SGD}\,T}\right)^{1/2}\sigma^2_{\rm SGD}\right) = \mathcal{O}\left(\frac{LF_0}{T} + \left(\frac{LF_0\,\sigma^2_{\rm SGD}}{T}\right)^{1/2}\right).$$

- $\gamma = \left(\frac{F_0}{L\,\sigma_{\rm SGD}^2\,T}\right)^{1/2}$. Then, substituting this value into (7) we get

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}\left(\left(\frac{LF_0\,\sigma_{\rm SGD}^2}{T}\right)^{1/2} + L\left(\frac{F_0}{L\,\sigma_{\rm SGD}^2\,T}\right)^{1/2}\sigma_{\rm SGD}^2\right) = \mathcal{O}\left(\left(\frac{LF_0\,\sigma_{\rm SGD}^2}{T}\right)^{1/2}\right)$$
$$\leq \mathcal{O}\left(\frac{LF_0}{T} + \left(\frac{LF_0\,\sigma_{\rm SGD}^2}{T}\right)^{1/2}\right),$$

  where in the last inequality we added a positive term $\frac{LF_0}{T} \geq 0$.

In both cases it holds that

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}\left(\frac{LF_0}{T} + \sqrt{\frac{LF_0\,\sigma_{\rm SGD}^2}{T}}\right),$$

which completes the proof. □

We see that this bound assumes that we fix the number of iterations T of our method in advance and use it in our analysis. As the result, the tuned value of the stepsize (25) depends on T. This assumption is standard for the stochastic first-order methods (see, e.g., (Ghadimi & Lan, 2013)). However, there is another simple possibility, which requires to fix the target accuracy $\varepsilon > 0$. In some cases, it can be more preferable as it does not require to know T in advance. Let us present this alternative choice, which leads to the same global complexity bounds.

Corollary E.2 (SGD, fixed target accuracy)**.** *Let each of the functions f_i be L-smooth (Assumption 4.1). Let i_t in Algorithm (2) be chosen as in Example 3.1, i.e. $i_t \sim [n]$. Let $\varepsilon > 0$ be any fixed accuracy and set*

$$\gamma = \min\left\{\frac{1}{7}, \frac{\varepsilon}{\sigma_{\rm SGD}^2}\right\}\cdot\frac{1}{2L}. \tag{26}$$

Then, for any $T \geq 1$, we have the following guarantee for the average gradient norm:

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}\left(\frac{LF_0}{T}\cdot\max\{1, \frac{\sigma_{\rm SGD}^2}{\varepsilon}\} + \varepsilon\right), \tag{27}$$

where $F_0 = f(\mathbf{x}_0) - f^\star$, and $\mathcal{O}(\cdot)$ hides absolute numerical constants that do not depend on any of our parameters. As a direct consequence, in order to ensure $\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}(\varepsilon)$, it is enough to perform

$$T \;=\; LF_0\cdot\mathcal{O}\left(\frac{1}{\varepsilon} + \frac{\sigma_{\rm SGD}^2}{\varepsilon^2}\right)$$

iterations of the method.

Proof. Indeed, by our choice of the stepsize (26) we guarantee both $\gamma \leq \frac{1}{14L}$ and $\gamma \leq \frac{\varepsilon}{2L\,\sigma_{\rm SGD}^2}$. Hence, substituting the latter bound into (7), we obtain

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}\left(\frac{F_0}{\gamma T} + L\gamma\,\sigma_{\rm SGD}^2\right) \leq \mathcal{O}\left(\frac{F_0}{\gamma T} + \frac{\varepsilon}{2}\right) = \mathcal{O}\left(\frac{2LF_0}{T}\cdot\max\{7, \frac{\sigma_{\rm SGD}^2}{\varepsilon}\} + \frac{\varepsilon}{2}\right),$$

which gives the required bound (27). □

E.2. Incremental Gradients, Single Shuffle, and Random Shuffle

In this section, we present the details on the stepsize tuning for IG, SS, and RR algorithms. Note that using Theorem 5.1 and estimating $\sigma_\tau^2 \leq 4\min\{\tau, n\} n\, \sigma_{\rm SGD}^2$ for all the three methods (see Table 2), we have the following bound for the average gradient norm:

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}\left(\frac{F_0}{\gamma T} + L\gamma\min\{1, L\gamma n\} n\, \sigma_{\rm SGD}^2\right), \tag{28}$$

valid for any $0 < \gamma \leq \frac{1}{14L} \leq \frac{1}{8\sqrt{3}L}$.

Now, for simplicity, let us consider first the following direct consequence of (28) (compare with (7)):

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}\left(\frac{F_0}{\gamma T} + L^2\gamma^2 n^2\, \sigma_{\rm SGD}^2\right). \tag{29}$$

As we demonstrate, this bound already leads to the better convergence rate of the Incremental Gradient and Single Shuffle methods, which improves upon the previous works. Indeed, analogously to Corollary E.1 for SGD, let us fix the number of iterations $T > 0$ for our method and minimize the right hand side of (29) over $0 < \gamma \leq \frac{1}{14L}$. Note that this is a univariate convex function with respect to γ, and its minimizer is attained at (compare with the stepsize rule (25) for SGD):

$$\gamma = \min\left\{\frac{1}{14L}, \left(\frac{F_0}{2L^2 n^2\, \sigma_{\rm SGD}^2\, T}\right)^{1/3}\right\}, \tag{30}$$

and the corresponding bound for the average gradient norm becomes

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}\left(\frac{LF_0}{T} + \left[\frac{LF_0 n\sigma_{\rm SGD}}{T}\right]^{2/3}\right).$$

Note that this rate already improves upon the previously known ones for IG and SS. However, we can make it even better by considering both arguments of the minimum in (28). We formulate the following precise statement.

Corollary E.3 (IG, SS and RR, fixed number of iterations)**.** *Let each of the functions f_i be L-smooth (Assumption 4.1). Let i_t in Algorithm (2) be chosen as in Example 3.2, 3.3, or 3.4, implementing either IG, SS or RR strategies. Let $T \geq 1$ be a fixed number of iterations. Then, after running Algorithm (2) for T iterations with the stepsize*

$$\gamma = \begin{cases} \min\left\{\left(\frac{F_0}{Ln\sigma^2 T}\right)^{\frac{1}{2}}, \frac{1}{14L}\right\} & \text{if } \left(\frac{Ln\sigma^2 F_0}{T}\right)^{\frac{1}{2}} \leq \left(\frac{F_0 Ln\sigma}{T}\right)^{\frac{2}{3}} \\ \min\left\{\left(\frac{F_0}{L^2 n^2 \sigma^2 T}\right)^{\frac{1}{3}}, \frac{1}{14L}\right\} & \text{otherwise} \end{cases}$$

the average gradient norm is bounded as follows:

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\|\nabla f(\mathbf{x}_t)\|^2 \leq \mathcal{O}\left(\frac{F_0 L}{T} + \min\left\{\left(\frac{Ln\sigma^2 F_0}{T}\right)^{\frac{1}{2}}, \left(\frac{F_0 Ln\sigma}{T}\right)^{\frac{2}{3}}\right\}\right)$$

where $F_0 = f(\mathbf{x}_0) - f^\star$, and $\mathcal{O}(\cdot)$ hides absolute numerical constants that do not depend on our parameters.

Proof. First, according to (8),

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\|\nabla f(\mathbf{x}_t)\|^2 \leq \mathcal{O}\left(\frac{F_0}{\gamma T} + L\gamma\min\{1, L\gamma n\} n\sigma^2\right),$$

where $0 < \gamma \leq \frac{1}{14L}$ is arbitrary. This means that both hold

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\|\nabla f(\mathbf{x}_t)\|^2 \leq \mathcal{O}\left(\frac{F_0}{\gamma T} + L\gamma n\sigma^2\right) \qquad \text{and} \qquad \frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\|\nabla f(\mathbf{x}_t)\|^2 \leq \mathcal{O}\left(\frac{F_0}{\gamma T} + L^2\gamma^2 n^2\sigma^2\right). \tag{a}$$

We will further tune the stepsize separately for each of these inequalities, and will show how to combine them in order to get the final convergence rate.

For inequality (a, left) we use Lemma 17 from (Koloskova et al., 2020). Below we repeat the calculations from this lemma for clarity. We take the stepsize $\gamma = \min\left\{\left(\frac{F_0}{Ln\sigma^2T}\right)^{\frac{1}{2}}, \frac{1}{14L}\right\} \leq \frac{1}{14L}$. Then we consider the two cases

- $\gamma = \frac{1}{14L}$, then $\gamma \leq \left(\frac{F_0}{Ln\sigma^2T}\right)^{\frac{1}{2}}$. We substitute our choice of γ in (a, left), and get

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}||\nabla f(\mathbf{x}_t)||^2 \leq \mathcal{O}\left(\frac{14F_0L}{T} + L\left(\frac{F_0}{Ln\sigma^2T}\right)^{\frac{1}{2}}n\sigma^2\right) = \mathcal{O}\left(\frac{F_0L}{T} + \left(\frac{F_0Ln\sigma^2}{T}\right)^{\frac{1}{2}}\right),$$

  where in the first term we used $\gamma = \frac{1}{12L}$, and in the second term we used that $\gamma \leq \left(\frac{F_0}{Ln\sigma^2(T+1)}\right)^{\frac{1}{2}}$.

- $\gamma = \left(\frac{F_0}{Ln\sigma^2T}\right)^{\frac{1}{2}}$. We substitute our choice of γ in (a, left), and get

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}||\nabla f(\mathbf{x}_t)||^2 \leq \mathcal{O}\left(\left(\frac{Ln\sigma^2F_0}{T}\right)^{\frac{1}{2}} + \left(\frac{Ln\sigma^2F_0}{T}\right)^{\frac{1}{2}}\right) \leq \mathcal{O}\left(\frac{F_0L}{T} + \left(\frac{Ln\sigma^2F_0}{T}\right)^{\frac{1}{2}}\right)$$

  where the last inequality is because $\frac{F_0L}{T} \geq 0$.

Therefore, with our choice of stepsize, in both cases we get that

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}||\nabla f(\mathbf{x}_t)||^2 \leq \mathcal{O}\left(\frac{F_0L}{T} + \left(\frac{Ln\sigma^2F_0}{T}\right)^{\frac{1}{2}}\right) \tag{b}$$

For inequality (a, right) we take the stepsize $\gamma = \min\left\{\left(\frac{F_0}{L^2n^2\sigma^2T}\right)^{\frac{1}{3}}, \frac{1}{14L}\right\} \leq \frac{1}{14L}$. Then we consider the two cases:

- $\gamma = \frac{1}{14L}$. Then $\gamma \leq \left(\frac{F_0}{L^2n^2\sigma^2T}\right)^{\frac{1}{3}}$, and substituting our choice of γ in (a, right) we get

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}||\nabla f(\mathbf{x}_t)||^2 \leq \mathcal{O}\left(\frac{14F_0L}{T} + L^2\left(\frac{F_0}{L^2n^2\sigma^2T}\right)^{\frac{2}{3}}n^2\sigma^2\right) \leq \mathcal{O}\left(\frac{F_0L}{T} + \left(\frac{F_0Ln\sigma}{T}\right)^{\frac{2}{3}}\right)$$

- $\gamma = \left(\frac{F_0}{L^2n^2\sigma^2T}\right)^{\frac{1}{3}}$. Substituting our choice of γ in (a, right) we get

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}||\nabla f(\mathbf{x}_t)||^2 \leq \mathcal{O}\left(\left(\frac{F_0Ln\sigma}{T}\right)^{\frac{2}{3}} + \left(\frac{F_0Ln\sigma}{T}\right)^{\frac{2}{3}}\right) \leq \mathcal{O}\left(\frac{F_0L}{T} + \left(\frac{F_0Ln\sigma}{T}\right)^{\frac{2}{3}}\right),$$

  where in the last inequality we used that $\frac{F_0L}{T} \geq 0$.

Therefore, with our choice of stepsize, in both cases we get that

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}||\nabla f(\mathbf{x}_t)||^2 \leq \mathcal{O}\left(\frac{F_0L}{T} + \left(\frac{F_0Ln\sigma}{T}\right)^{\frac{2}{3}}\right). \tag{c}$$

Combining convergence rates (b) and (c). We now explain how do we combine (b) and (c) in order to get the final convergence rate. We employ the following stepsize

$$\gamma = \begin{cases} \min\left\{\left(\frac{F_0}{Ln\sigma^2T}\right)^{\frac{1}{2}}, \frac{1}{14L}\right\} & \text{if } \left(\frac{Ln\sigma^2F_0}{T}\right)^{\frac{1}{2}} \leq \left(\frac{F_0Ln\sigma}{T}\right)^{\frac{2}{3}} \\ \min\left\{\left(\frac{F_0}{L^2n^2\sigma^2T}\right)^{\frac{1}{3}}, \frac{1}{14L}\right\} & \text{otherwise} \end{cases}$$

Therefore, we get the following resulting convergence rate

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}||\nabla f(\mathbf{x}_t)||^2 \leq \begin{cases} \mathcal{O}\left(\frac{F_0 L}{T} + \left(\frac{Ln\sigma^2 F_0}{T}\right)^{\frac{1}{2}}\right) & \text{if } \left(\frac{Ln\sigma^2 F_0}{T}\right)^{\frac{1}{2}} \leq \left(\frac{F_0 Ln\sigma}{T}\right)^{\frac{2}{3}} \\ \mathcal{O}\left(\frac{F_0 L}{T} + \left(\frac{F_0 Ln\sigma}{T}\right)^{\frac{2}{3}}\right) & \text{otherwise} \end{cases}$$

In other words,

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}||\nabla f(\mathbf{x}_t)||^2 \leq \mathcal{O}\left(\frac{F_0 L}{T} + \min\left\{\left(\frac{Ln\sigma^2 F_0}{T}\right)^{\frac{1}{2}}, \left(\frac{F_0 Ln\sigma}{T}\right)^{\frac{2}{3}}\right\}\right),$$

which completes the proof. □

Finally, let us consider an alternative approach, when we fix the target accuracy $\varepsilon > 0$ instead of the number of iterations, analogously to Corollary E.2 for SGD. This approach leads to the same complexity bounds for IG, SS, and RR as those ones from the previous Corollary E.3. However, it can be preferable if we do not want to fix the number of iterations T for our method in advance.

Corollary E.4 (IG, SS and RR, fixed target accuracy)**.** *Let each of the functions f_i be L-smooth (Assumption 4.1). Let i_t in Algorithm (2) be chosen as in Example 3.2, 3.3, or 3.4, implementing either IG, SS or RR strategies. Let $\varepsilon > 0$ be any fixed accuracy and set*

$$\gamma = \min\left\{\frac{1}{7}, \max\left[\frac{\varepsilon}{n\,\sigma_{\mathrm{SGD}}^2}, \sqrt{\frac{2\varepsilon}{n^2\,\sigma_{\mathrm{SGD}}^2}}\right]\right\}\cdot\frac{1}{2L}. \tag{31}$$

Then, for any $T \geq 1$, we have the following guarantee for the average gradient norm:

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}||\nabla f(\mathbf{x}_t)||^2 \leq \mathcal{O}\left(\frac{LF_0}{T}\cdot\max\left\{1, \min\left[\frac{n\,\sigma_{\mathrm{SGD}}^2}{\varepsilon}, \sqrt{\frac{n^2\,\sigma_{\mathrm{SGD}}^2}{\varepsilon}}\right]\right\} + \varepsilon\right)$$

where $F_0 = f(\mathbf{x}_0) - f^\star$, and $\mathcal{O}(\cdot)$ hides absolute numerical constants that do not depend on our parameters. As a direct consequence, in order to ensure $\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}(\varepsilon)$, it is enough to perform

$$T \;=\; LF_0\cdot\mathcal{O}\left(\frac{1}{\varepsilon} + \min\left\{\frac{n\,\sigma_{\mathrm{SGD}}^2}{\varepsilon^2}, \frac{n\sigma_{\mathrm{SGD}}}{\varepsilon^{3/2}}\right\}\right) \tag{32}$$

iterations of the method.

Proof. Indeed, by our choice of the stepsize (31) we guarantee that both $\gamma \leq \frac{1}{14L}$ and $\gamma \leq \frac{1}{2L}\max\left[\frac{\varepsilon}{n\,\sigma_{\mathrm{SGD}}^2}, \sqrt{\frac{2\varepsilon}{n^2\,\sigma_{\mathrm{SGD}}^2}}\right]$. The latter inequality ensures that

$$\min\{L\gamma, L^2\gamma^2 n\} \leq \frac{\varepsilon}{2n\,\sigma_{\mathrm{SGD}}^2}. \tag{33}$$

Hence, substituting this estimate into (28), we get

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}||\nabla f(\mathbf{x}_t)||^2 \overset{(28)}{\leq} \mathcal{O}\left(\frac{F_0}{\gamma T} + L\gamma\min\{1, L\gamma n\}n\,\sigma_{\mathrm{SGD}}^2\right) \overset{(33)}{\leq} \mathcal{O}\left(\frac{F_0}{\gamma T} + \frac{\varepsilon}{2}\right).$$

Substituting the value (31) of γ completes the proof. □

Remark E.5. Note that complexity (32) that we established for IG, SS, and RR algorithms exactly corresponds to that one from Table 1.