Shuffle SGD: convergence rate analysis for convex and non-convex problems

Aishwarya Hiwrale
Javed Akhtar
Tom Jacobs
Department of Computer Science, University of Saarland, Germany

***Abstract*—First order sub-gradient methods are a cornerstone in machine learning. In this paper we consider a group of methods, which are variants of the (stochastic) gradient descent algorithm. It has been shown that members of this group have distinct convergence rates. Making particular members more favorable, leading to shorter training and saving computation. The convergence rates assume that the machine learning task is $L-$smooth and not necessarily convex. We identify these rates for particular machine learning tasks. We consider both a strongly convex and a convex task. The considered tasks are linear regression and logistic regression. In experiments it is shown that the identified convergence rates hold. Moreover, we observe that the strongly convex task enjoys faster convergence rates then the derived rate. The faster rates still respect the convergence rate ordering between the members of the group. This opens the door for studying local strongly convex or Polyak-Lojasiewicz functions.**

I. INTRODUCTION

- Hook: practice vs theory, sgd is theoretically tractable and easier to analyze, while in practice variants as single shuffle and random shuffle are used.
- Faster convergence saves compute
- Strongly vs non-strongly convex functions.
- it has been shown that for regression machine learning tasks the noise satisfies a decaying noise condition. Potentially it also satisfies the PL-inequality locally leading to different convergence properties than for just L-smooth loss functions.
- Therefore it is of importance to investigate the convergence speed in these cases as they might differ from the noise assumption in [1].
-

We investigate several variants of SGD and their convergence rate. The variants are plain SGD, single shuffle (SS), and random reshuffle (RR). Recently it has been shown for L-smooth objective functions of the form $f := \sum_i^n f_i$ that variants as SS and RR converge faster than SGD. The proof hinges on the introduction of sequence correlation Definition 4.2 in [1]. This leads to the noise strength parameter σ_τ. In contrast to the conventional noise assumption for SGD:

$$\sigma^2_{\text{SGD}} := \sup_{w\in\mathbb{R}^d} \mathbb{E}_i||\nabla f_i(w) - \nabla f(w)||^2, \tag{1}$$

σ_τ allows us to deal with the correlation that inevitably happens in case of SS and RR.

In addition, under the additional assumption of the PL-inequality other results where known already before [2], [3], [4]. The Polyak-Lojasiewicz inequality or PL-inequality is given by

$$|\nabla g(w)|^2 \geq \Lambda g(w) \qquad \forall w \in \mathbb{R}^d, \tag{2}$$

where $\Lambda > 0$ is a positive constant. In [5], [6] it is proposed that regression tasks in ML can satisfy the PL-inequality locally. Furthermore, the noise also satisfies a growth condition instead of (1):

$$\mathbb{E}_i||\nabla f_i(w) - \nabla f(w)||^2 \leq M_{\text{SGD}} f(w) \qquad \forall w \in \mathbb{R}^d \tag{3}$$

which is similar to the assumptions made in [7]. Nevertheless it is an open question if a sequential correlation definition analogue can be made for (3). Therefore we investigate with on the hand of two machine learning tasks: strongly convex (i.e. satisfies the PL-inequality) and convex f_i. We investigate the convergence rate and if there is a difference between strongly convex and convex functions.

We first give the experimental details in Section II. Next, the results are presented in Section III Furthermore we propose an analogue definition of sequential correlation for (3) and discuss the results in Section IV. Finally, in Section V we summarize our findings and present our conclusions.

Our main contributions are:

- Investigating the convergence rate with two different experiments linear and logistic regression. This confirms the insights from [1].
- Shedding light on the difference in noise structure between classification and regression tasks.
- Conjecturing an analogue for sequential correlation capturing the alternative noise structure in (3).

II. TASKS AND EXPERIMENTAL DETAILS

In this section we provide the details of each algorithm. We describe each shortly and provide their convergence rate. The convergence rate depends on multiple parameters. A task specific parameter is the Lipschitz constant L. For the

Consider a loss function $f : \mathbb{R}^d \to \mathbb{R}$ of the finite sum form $\sum_{i=1}^n f_i$. We minimize this loss function with the following algorithm

$$w_{t+1} = w_t - \gamma\nabla f_{i_t}(w_t) \tag{4}$$

where $\gamma > 0$ is the learning rate and i_t is random or deterministic index. Several variants of SGD are covered by

Algorithm	σ_τ^2 upperbound
SGD	$\tau\sigma_{\text{SGD}}^2$
SS	$\min\{\tau, n\}n\sigma_{\text{SGD}}^2$
RR	$4\min\{\tau, n\}n\sigma_{\text{SGD}}^2$

TABLE I: Upperbound on σ_τ^2.

this formulation. The variants we explore in this work are standard SGD (SGD), single shuffle (SS), random reshuffle (RR). The main difference can be described by how i_t is sampled. SGD samples at every step i_t uniformly at random. SS chooses a fixed order (permutation) and goes trough all indices $1, \dots, n$. The order is chosen at random. RR similarly to SS goes trough all indices, nevertheless, the order changes every time.

In case all f_i are L-smooth functions a convergence rate has been derived. According to Theorem 5.1 in [1] the convergence rate is

$$\mathcal{O}\left(\frac{F_0}{\gamma T} + L^2\gamma^2\sigma_\tau^2\right). \qquad (5)$$

where $F_0 = f(w_0) - f^*$ and σ_τ the noise strength. In Table I we present the bounds on σ_τ^2 in terms of σ_{SGD}^2 for each algorithm. Note that previous known bounds can be recovered and improved.

a) Linear Regression: We consider a problem that satisfies the PL-inequality, namely, linear regression. Consider a dataset $\{x_i, y_i\}_{i=1}^n$ with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. Let $f_i(w) := \frac{1}{2}\left(w^T x_i - y_i\right)^2$. Then each f_i is L_i-smooth with $L_i = \lambda_{\text{max}}\left(x_i x_i^T\right)$, where λ_{max} means largest eigen value. This follows from computing the Hessian. Thus Theorem 5.1 in [1] applies with $L = \max_i L_i$. Λ in the PL-inequality for each f_i is $\lambda_{\text{min}}(x_i x_i^T)$.

b) Logistic Regression: Similar as in [1] we consider logistic regression which is convex but does not satisfy the PL-inequality. Consider a dataset $\{x_i, y_i\}_{i=1}^n$ with $x_i \in \mathbb{R}^d$ and $y_i \in \{\pm 1\}$. Let $f_i(w) := \log\left(1 + e^{-y_i w^T x_i}\right)$. Then each f_i is L_i-smooth with $L_i = \frac{1}{4}\lambda_{\text{max}}\left(x_i x_i^T\right)$. This follows from computing the Hessian for each f_i and using the following elementary inequality

$$\frac{e^z}{\left(1 + e^z\right)^2} \leq \frac{1}{4} \qquad \forall z \in \mathbb{R}.$$

Thus Theorem 5.1 in [1] applies with $L = \max_i L_i$.

c) Data sets: For all experiments we run 5 seeds and plot the errorbars. In each case we initialize in the same way, we start training from $w_0 = 0$.

For linear regression we use an artificially created data set based on the experimental setup in [8]. We generate a sparse ground truth w^* such that $||w||_{L0} = 5$. The setup is chosen for its easy check for generalizability. We compare the found solution with the ground truth w^* by measuring the euclidean distance. This gives another dimension to our analysis.

For logistic regression we consider two toy data sets *australian* from [9] and *breast_cancer* from [10]. The first data set is used as a benchmark to compare with [1]. The second data set is an extension.

III. Results

In this section we present our experimental findings. We start wit linear regression and move on finish with logistic regression. We present for all experiments multiple constant learning rates $\gamma > 0$. In addition we compute the Lipschitz coefficient for each problem as derived in the previous section.

a) Linear Regression: For the sparse linear regression we have a Lipschitz constant of 111.1. In Figure 1 we present the trajectory of the roling average gradient norm squared as in Theorem 5.1 in [1]. This is done for 3 different learning rates.

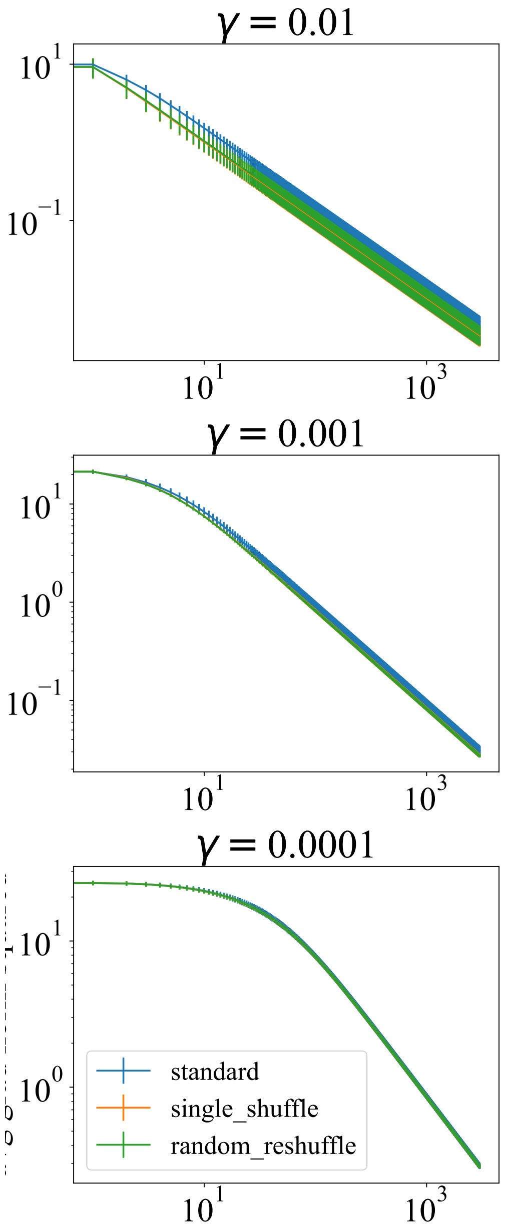

Fig. 1: Linear regression with sparse ground truth

b) Logistic Regression: We consider two data sets for logistic regression as mentioned in the previous section. For the two data sets we compute the Lipschitz constant. In case of the *Australian* dataset the Lipschitz coefficient is 3.1 and for the *Breast_cancer* dataset the coefficient is 5.5. The results are presented in Figures **??** and 2.

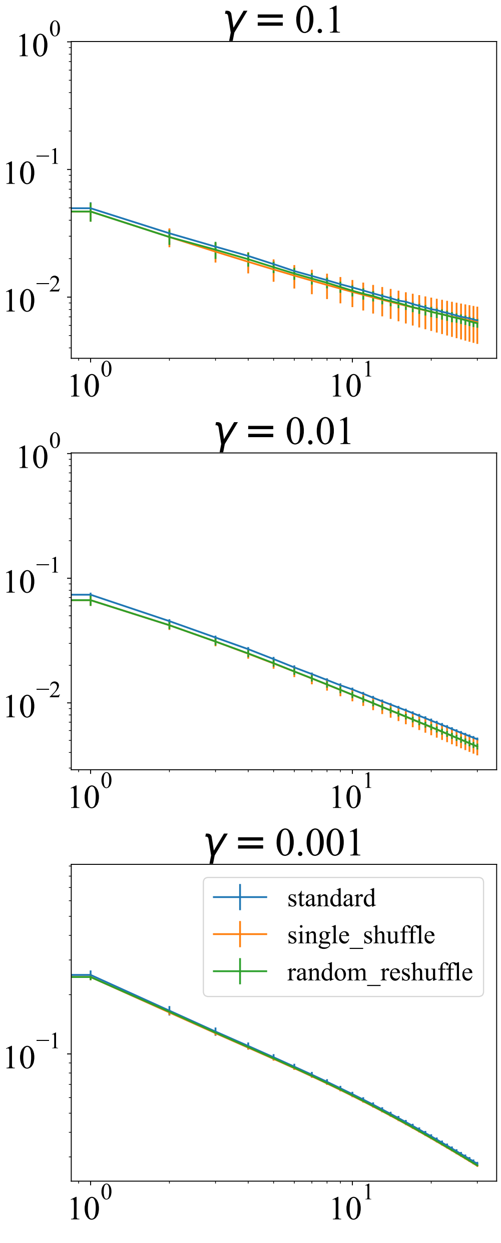

Fig. 2: Logistic regression on the *breast_cancer* dataset.

- Track rates average over initializations, plot theory convergence rate in there as well
- generalization performance of the algorithms is another important factor so also make a test set
- Total 4 plots

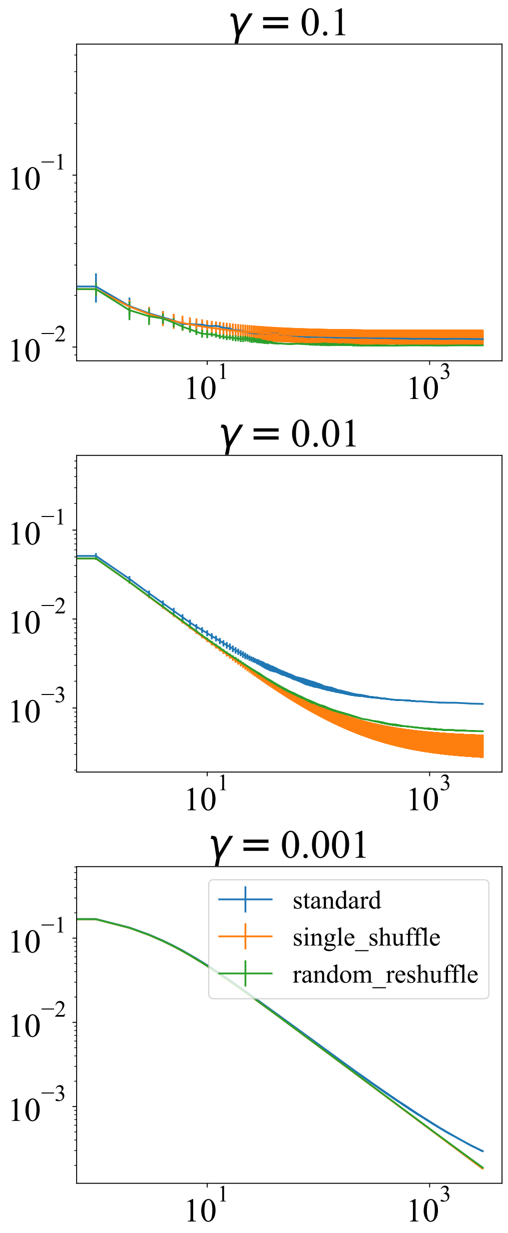

Fig. 3: Logistic regression on the *australian* dataset.

IV. Discussion

We observe that indeed that SS and RR outperform SGD in both experiments.

- Faster convergence for linear regression as expected
- Still ordering persists
- L-smoothness is based on the maximum of the L_i but in practice larger learning rates also work fine.
- Add a bit more about conjecture

As mentioned in the introduction for regression tasks a generalization sequential correlation as in Definition 4.1 [1] is needed for noise of the type (3). The natural generalization of this is

$$\max_{k=0,\dots,\lfloor \frac{T}{\tau} \rfloor, j=0,\dots,\tau-1} \mathbb{E}\left[\phi_{k\tau+j}(x) | i_0, \dots, i_{k\tau-1}\right] \leq M_\tau f(x) \tag{6}$$

with $\phi_{k\tau+j}(x)$ being defined as

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

Nevertheless, this generalization might make the proof more cumbersome. Because now the right hand side in (6) depends on x. Note that the same bounds as in Table I can be derived for M_τ in terms of M_{SGD}.

- Describe differences between the algorithms i.e. rates and generalization
- stongly convex vs non-convex
- propose new noise condition
- informally present difficulties in the proof

V. Summary

VI. The Structure of a Paper

Scientific papers usually begin with the description of the problem, justifying why the problem is interesting. Most importantly, it argues that the problem is still unsolved, or that the current solutions are unsatisfactory. This leads to the main gist of the paper, which is "the idea". The authors then show evidence, using derivations or experiments, that the idea works. Since science does not occur in a vacuum, a proper comparison to the current state of the art is often part of the results. Following these ideas, papers usually have the following structure:

Abstract
: Short description of the whole paper, to help the reader decide whether to read it.

Introduction
: Describe your problem and state your contributions.

Models and Methods
: Describe your idea and how it was implemented to solve the problem. Survey the related work, giving credit where credit is due.

Results
: Show evidence to support your claims made in the introduction.

Discussion
: Discuss the strengths and weaknesses of your approach, based on the results. Point out the implications of your novel idea on the application concerned.

Summary
: Summarize your contributions in light of the new results.

VII. Tips for Good Writing

The ideas for good writing have come from [11], [12], [13].

A. Getting Help

One should try to get a draft read by as many friendly people as possible. And remember to treat your test readers with respect. If they are unable to understand something in your paper, then it is highly likely that your reviewers will not understand it either. Therefore, do not be defensive about the criticisms you get, but use it as an opportunity to improve the paper. Before your submit your friends to the pain of reading your draft, please *use a spell checker*.

B. Abstract

The abstract should really be written last, along with the title of the paper. The four points that should be covered [12]:

1) State the problem.
2) Say why it is an interesting problem.
3) Say what your solution achieves.
4) Say what follows from your solution.

C. Figures and Tables

Use examples and illustrations to clarify ideas and results. For example, by comparing Figure 4 and Figure 5, we can see the two different situations where Fourier and wavelet basis perform well.

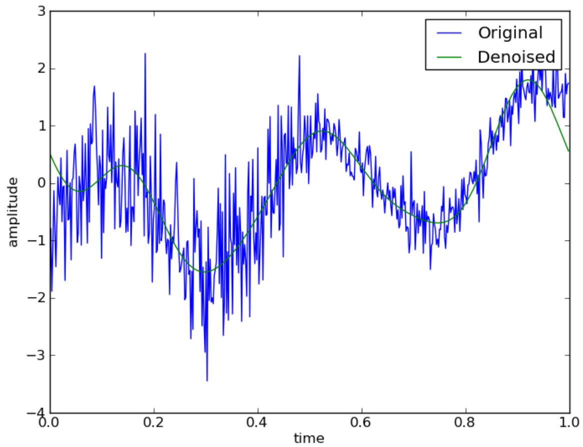

Fig. 4: Signal compression and denoising using the Fourier basis.

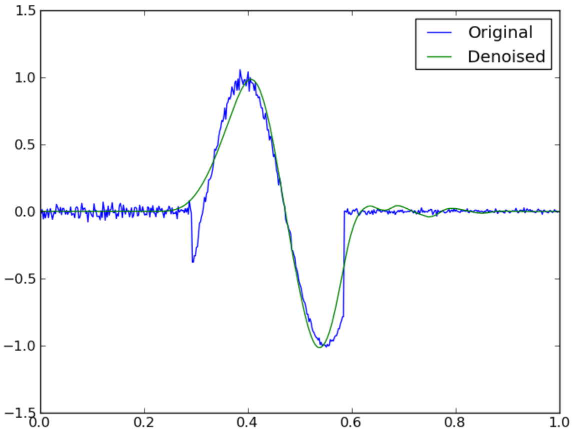

Fig. 5: Signal compression and denoising using the Daubechies wavelet basis.

D. Models and Methods

The models and methods section should describe what was done to answer the research question, describe how it was done, justify the experimental design, and explain how the results were analyzed.

The model refers to the underlying mathematical model or structure which you use to describe your problem, or that your solution is based on. The methods on the other hand, are the algorithms used to solve the problem. In some cases, the suggested method directly solves the problem, without having it stated in terms of an underlying model. Generally though it is a better practice to have the model figured out and stated clearly, rather than presenting a method without specifying the model. In this case, the method can be more easily evaluated in the task of fitting the given data to the underlying model.

The methods part of this section, is not a step-by-step, directive, protocol as you might see in your lab manual, but detailed enough such that an interested reader can reproduce your work [13], [14].

The methods section of a research paper provides the information by which a study's validity is judged. Therefore, it requires a clear and precise description of how an experiment was done, and the rationale for why specific experimental procedures were chosen. It is usually helpful to structure the methods section by [15]:

1) Layout the model you used to describe the problem or the solution.
2) Describing the algorithms used in the study, briefly including details such as hyperparameter values (e.g. thresholds), and preprocessing steps (e.g. normalizing the data to have mean value of zero).
3) Explaining how the materials were prepared, for example the images used and their resolution.
4) Describing the research protocol, for example which examples were used for estimating the parameters (training) and which were used for computing performance.
5) Explaining how measurements were made and what calculations were performed. Do not reproduce the full source code in the paper, but explain the key steps.

E. Results

Organize the results section based on the sequence of table and figures you include. Prepare the tables and figures as soon as all the data are analyzed and arrange them in the sequence that best presents your findings in a logical way. A good strategy is to note, on a draft of each table or figure, the one or two key results you want to address in the text portion of the results. The information from the figures is summarized in Table II.

When reporting computational or measurement results, always report the mean (average value) along with a measure of variability (standard deviation(s) or standard error of the mean).

VIII. Tips for Good Software

There is a lot of literature (for example [16] and [17]) on how to write software. It is not the intention of this section to replace software engineering courses. However, in the interests of reproducible research [18], there are a few guidelines to make your reader happy:

- Have a `README` file that (at least) describes what your software does, and which commands to run to obtain results. Also mention anything special that needs to be set up, such as toolboxes[1].
- A list of authors and contributors can be included in a file called `AUTHORS`, acknowledging any help that you may have obtained. For small projects, this information is often also included in the `README`.

[1] For those who are particularly interested, other common structures can be found at http://en.wikipedia.org/wiki/README and http://www.gnu.org/software/womb/gnits/.

Basis	Support	Suitable signals	Unsuitable signals
Fourier	global	sine like	localized
wavelet	local	localized	sine like

TABLE II: Characteristics of Fourier and wavelet basis.

- Use meaningful filenames, and not `temp1.py`, `temp2.py`.
- Document your code. Each file should at least have a short description about its reason for existence. Non obvious steps in the code should be commented. Functions arguments and return values should be described.
- Describe how the results presented in your paper can be reproduced.

A. *LaTeX Primer*

LaTeX is one of the most commonly used document preparation systems for scientific journals and conferences. It is based on the idea that authors should be able to focus on the content of what they are writing without being distracted by its visual presentation. The source of this file can be used as a starting point for how to use the different commands in LaTeX. We are using an IEEE style for this course.

1) Installation: There are various different packages available for processing LaTeX documents. See our webpage for more links for getting started.

2) Compiling LaTeX: Your directory should contain at least 4 files, in addition to image files. Images should ideally be `.pdf` format (or `.png`).

3) Equations: There are three types of equations available: inline equations, for example $y = mx + c$, which appear in the text, unnumbered equations

$$y = mx + c,$$

which are presented on a line on its own, and numbered equations

$$y = mx + c \tag{7}$$

which you can refer to at a later point (Equation (7)).

4) Tables and Figures: Tables and figures are "floating" objects, which means that the text can flow around it. Note that `figure*` and `table*` cause the corresponding figure or table to span both columns.

IX. Summary

The aim of a scientific paper is to convey the idea or discovery of the researcher to the minds of the readers. The associated software package provides the relevant details, which are often only briefly explained in the paper, such that the research can be reproduced. To write good papers, identify your key idea, make your contributions explicit, and use examples and illustrations to describe the problems and solutions.

Acknowledgements

The author thanks Christian Sigg for his careful reading and helpful suggestions.

References

[1] A. Koloskova, N. Doikov, S. U. Stich, and M. Jaggi, "On convergence of incremental gradient for non-convex smooth functions," 2024. [Online]. Available: https://arxiv.org/abs/2305.19259

[2] K. Ahn, C. Yun, and S. Sra, "Sgd with shuffling: optimal rates without component convexity and large epoch requirements," 2020. [Online]. Available: https://arxiv.org/abs/2006.06946

[3] L. M. Nguyen, Q. Tran-Dinh, D. T. Phan, P. H. Nguyen, and M. van Dijk, "A unified convergence analysis for shuffling-type gradient methods," 2021. [Online]. Available: https://arxiv.org/abs/2002.08246

[4] J. Cha, J. Lee, and C. Yun, "Tighter lower bounds for shuffling sgd: Random permutations and beyond," 2023. [Online]. Available: https://arxiv.org/abs/2303.07160

[5] S. Wojtowytsch, "Stochastic gradient descent with noise of machine learning type. part i: Discrete time analysis," 2021. [Online]. Available: https://arxiv.org/abs/2105.01650

[6] B. Gess and S. Kassing, "Convergence rates for momentum stochastic gradient descent with noise of machine learning type," 2023. [Online]. Available: https://arxiv.org/abs/2302.03550

[7] A. Mohtashami, S. Stich, and M. Jaggi, "Characterizing finding good data orderings for fast convergence of sequential gradient methods," 2022. [Online]. Available: https://arxiv.org/abs/2202.01838

[8] S. Pesme, L. Pillaud-Vivien, and N. Flammarion, "Implicit bias of sgd for diagonal linear networks: a provable benefit of stochasticity," 2021. [Online]. Available: https://arxiv.org/abs/2106.09524

[9] R. Quinlan, "Statlog (Australian Credit Approval)," UCI Machine Learning Repository, DOI: https://doi.org/10.24432/C59012.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[11] Editorial, "Scientific writing 101," *Nature Structural & Molecular Biology*, vol. 17, p. 139, 2010.

[12] S. P. Jones, "How to write a great research paper," 2008, microsoft Research Cambridge.

[13] G. Anderson, "How to write a paper in scientific journal style and format," 2004, http://abacus.bates.edu/ ganderso/biology/resources/writing/HTWtoc.html.

[14] J. B. Buckheit and D. L. Donoho, "Wavelab and reproducible research," Stanford University, Tech. Rep., 2009.

[15] R. H. Kallet, "How to write the methods section of a research paper," *Respiratory Care*, vol. 49, no. 10, pp. 1229–1232, 2004.

[16] A. Hunt and D. Thomas, *The Pragmatic Programmer*. Addison Wesley, 1999.

[17] J. Spolsky, *Joel on Software: And on Diverse & Occasionally Related Matters That Will Prove of Interest etc..: And on Diverse and Occasionally Related Matters ... or Ill-Luck, Work with Them in Some Capacity*. APRESS, 2004.

[18] M. Schwab, M. Karrenbach, and J. Claerbout, "Making scientific computations reproducible," *Computing in Science and Engg.*, vol. 2, no. 6, pp. 61–67, 2000.