MSL: An Adaptive Momentum-based Stochastic Line-search Framework

Chen Fan FANCHEN3@OUTLOOK.COM
University of British Columbia, Canada
Sharan Vaswani VASWANI.SHARAN@GMAIL.COM
Simon Fraser University, Canada
Christos Thrampoulidis CTHRAMPO@ECE.UBC.CA
University of British Columbia, Canada
Mark Schmidt SCHMIDTM@CS.UBC.CA
Canada CIFAR AI Chair (Amii)
University of British Columbia, Canada

Abstract

Various adaptive step sizes have been proposed recently to reduce the amount of tedious manual tuning. A popular example is back-tracking line-search based on a stochastic Armijo condition. But the success of this strategy relies crucially on the search direction being a descent direction. Importantly, this condition is violated by both SGD with momentum (SGDM) and Adam, which are common choices in deep-net training. Adaptively choosing the step size in this setting is thus non-trivial and less explored despite its practical relevance. In this work, we propose two frameworks, namely, momentum correction and restart, that allow the use of stochastic line-search in conjunction with a generalized Armijo condition, and apply them to both SGDM and Adam. We empirically verify that the proposed algorithms are robust to the choice of the momentum parameter and other hyperparameters.

1. Introduction

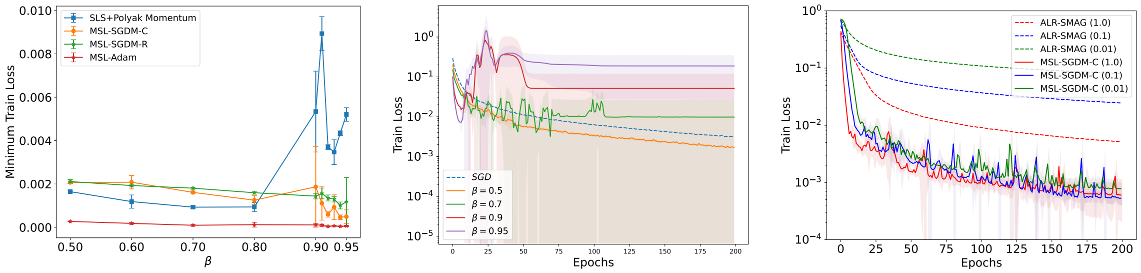
Many machine learning problems can be formulated as minimizing a finite-sum objective, i.e.:

$$\min\left\{ f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) \right\},$$

where n is the total number of data. A standard approach for solving such problems is stochastic gradient descent (SGD):

$$x_{k+1} = x_k - \eta_k \nabla f_{S_k}(x_k),$$

in which a random batch of samples S_k is drawn at each iteration k. The step size of SGD (i.e. η_k) is important for achieving stable and fast convergence of the algorithm [1]. This lead to recent research on finding a good step size policy for SGD without extensive tuning. Such approaches commonly take inspirations from the deterministic scenario [8, 18]. For example, Loizou et al. [8] extended the classic Polyak step size to the stochastic setting (SPS) when a lower bound $f^*_{S_k}$ on the function value f_{S_k} is available. Vaswani et al. [18] introduced a stochastic version of the Armijo



(a) Minimum train loss (b) Different βs (c) Different η_0s

Figure 1: Results of binary classification on a synthetic dataset using logistic loss (see Sec 3). (a) Minimum train loss against the momentum parameter β. SLS+Polyak Momentum refers to (2) where η_k is found by SLS [18] (see Sec 2 for MSL-based algorithms); (b) train loss of SGD with Polyak momentum against epochs for different βs (see (2)) with the step size being determined by SLS [18]; (c) similar plot as (b) with different η_0s in ALG-SMAG (where $\eta_k = \min\{\frac{f_{S_k} - f^*_{S_k}}{c\|d_k\|^2}, \eta_0\}$) and search initializations $\tilde{\eta}_0$s in MSL-SGDM-C (see Sec 3). Note that η_0 and $\tilde{\eta}_0$ are in brackets given in the legend.

line-search condition (SLS) based on the gradient ∇f_{S_k}. These adaptive step sizes have been shown to work well for over-parameterized models that can interpolate [8, 18].

In practice, momentum is frequently added to SGD to further improve its performance [17]. Instead of taking a step along the stochastic gradient direction ∇f_{S_k}, SGD with momentum modifies the update to:

$$x_{k+1} = x_k - \eta_k d_k, \tag{1}$$

where d_k typically includes some previous gradient information. One such variant sets $d_k = (1 - \beta)\nabla f_{S_k}(x_k) + \beta d_{k-1}$ where $\beta \in (0, 1)$ (known as SGDM [7, 17]). When Polyak momentum is combined with SGD [15], the resulting update is

$$x_{k+1} = x_k - \eta_k \nabla f_{S_k}(x_k) + \beta(x_k - x_{k-1}). \tag{2}$$

The momentum parameter β in the d_k of SGDM or (2) complicates the selection of a suitable step size. Directly applying existing approaches such as the stochastic line search (SLS) of Vaswani et al. [18] neglects β when adjusting the step size η_k. Consequently, as demonstrated in Figure 1(a) and Figure 1(b), this approach is not robust to the choice of momentum parameter β, and a large β can potentially lead to divergence. In another work, Wang et al. [20] proposed to use Polyak step size of the form $\eta_k = \min\{\frac{f_{S_k} - f^*_{S_k}}{c\|d_k\|^2}, \eta_0\}$. However, as we observe in Figure 1(c), this step size is highly sensitive to the choice of η_0. Thus, in the presence of momentum, the question of how to design an adaptive step size strategy that is robust to the momentum parameter as well as other choices of hyperparameters is not fully addressed. In this work, we propose a new line-search framework that can be applied to both SGDM and Adam, building upon a stochastic variant of the generalized Armijo condition. We discuss our approach in details in the next section.

2. Summary of Contributions

In this section, we first discuss the challenges in extending the generalized Armijo condition to the stochastic setting. To this end, we come up with a strategy that makes modifications to the momentum direction and apply it to SGDM. Furthermore, we propose another computationally-favorable approach that restarts the momentum for both Adam and SGDM cases.

2.1. Technical Challenge and Momentum Correction

We introduce the following generalized Armijo condition for momentum-based stochastic line-search (MSL)

$$f_{S_k}(x_k - \eta_k d_k) \leq f_{S_k}(x_k) - c\eta_k \langle \nabla f_{S_k}(x_k), d_k \rangle, \tag{3}$$

where d_k is some descent direction, i.e. $\langle \nabla f_{S_k}(x_k), d_k \rangle > 0$. In the special case of $d_k = \nabla f_{S_k}(x_k)$, (3) reduces to the search condition in SLS [18], for which descent on f_{S_k} is guaranteed when the step size η_k satisfies (3). However, this does not always hold true for other forms of d_k that involve momentum. For example, when $d_k = (1-\beta)\nabla f_{S_k}(x_k) + \beta d_{k-1}$ in the case of SGDM, we no longer have the guarantee that the inner product $\langle \nabla f_{S_k}(x_k), d_k \rangle$ is always positive. This is because the inner product $\langle \nabla f_{S_k}(x_k), d_{k-1} \rangle$ can be negative. We also empirically observe this as shown by the points labeled with -1 in Figure 2 (left). To resolve this challenge, we consider two situations. In the case where $\langle \nabla f_{S_k}(x_k), d_k \rangle > 0$, we use it directly in (3); otherwise, we first damp β to $\tilde{\beta}$ (e.g. half β each time) until the inner product $\langle \nabla f_{S_k}(x_k), d_k \rangle$ with d_k computed using $\tilde{\beta}$ (denoted as $\tilde{d}_k$) is positive, and use this inner product together with $\tilde{d}_k$ to find the step size and perform the update. By doing such momentum corrections, we are guaranteed a descent direction d_k where the step size found by (3) leads to a decrease in f_{S_k}. We name this algorithm as MSL-SGDM-C. As shown in Figure 2, damped momentum parameter $\tilde{\beta}$ ensures that all inner products $\langle \nabla f_{S_k}(x_k), \tilde{d}_k \rangle$ are positive.

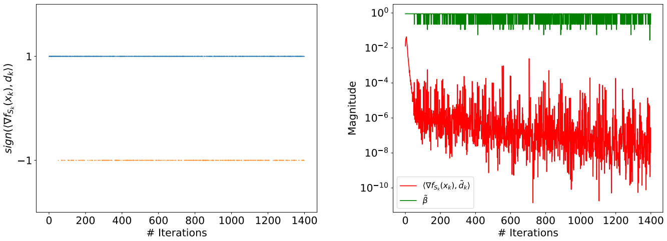
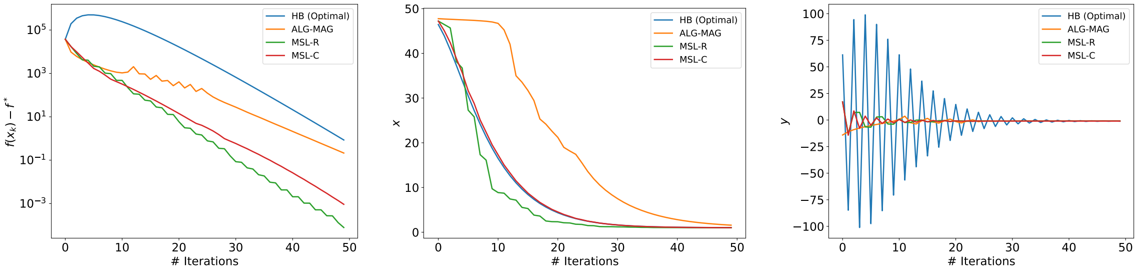


Figure 2: Results of binary classification on a synthetic dataset using our algorithm MSL-SGDM-C and logistic loss (see Sec 2.1 and 3). Left: sign($\langle \nabla f_{S_k}(x_k), d_k \rangle$) against iterations; right: damped momentum parameter $\tilde{\beta}$ and $\langle \nabla f_{S_k}(x_k), \tilde{d}_k \rangle$ against iterations.

2.2. Momentum Restart and its Extension to Adam

Besides making corrections to the search direction d_k, another approach is to restart d_k when $\langle \nabla f_{S_k}(x_k), d_k \rangle < 0$, i.e. set $d_k = (1-\beta)\nabla f_{S_k}(x_k) + \beta d_0$ in the case of SGDM, which reduces to $(1-\beta)\nabla f_{S_k}(x_k)$ given the initialization d_0 is chosen to be 0. We note that a similar restart strategy has been used in non-linear conjugate gradient methods [11, 14, 16]. Comparing to the momentum correction approach which requires checking the condition $\langle \nabla f_{S_k}(x_k), \tilde{d}_k \rangle > 0$ when damping the momentum parameter β, momentum restart avoids this while ensuring a descent on the



(a) Function value gap (b) x vs. # Iterations (c) y vs. # Iterations

Figure 3: Results on minimizing the 2-dimensional function $f(x, y) = \frac{1}{2}(x-1)^2 + \frac{\kappa}{2}(y+1)^2$. Experiments are adapted from Wang et al. [20] with the initial point $(x_0, y_0) = (48, -28)$, $\kappa = 100$, and $d_k = \nabla f_{S_k}(x_k) + \beta d_{k-1}$. The step size for ALG-MAG is $\eta_k = \frac{f(x_k) - f(x^*)}{\|d_k\|^2}$ [20], and for heavy-ball momentum (HB) optimal is $\eta_k = (1 + \sqrt{\beta^*})^2 / L$ where $\beta^* = (\sqrt{\kappa} - 1)^2 / (\sqrt{\kappa} + 1)^2$ [13]. Our methods are labelled with MSL-R and MSL-C, for momentum restart and corrections, respectively.

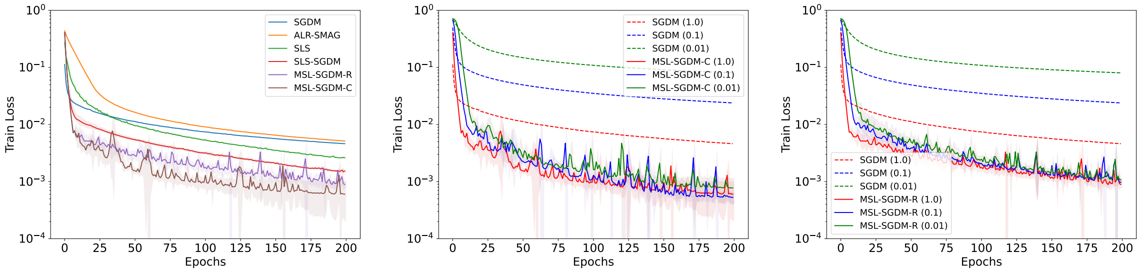
function f_{S_k} can be achieved. This is because both the search and update directions are proportional to ∇f_{S_k} when $\langle \nabla f_{S_k}(x_k), d_k \rangle < 0$.

Besides applying momentum restart to SGDM (MSL-SGDM-R), we can also apply it to Adam to set its step size. The Adam update is as follows:

$$m_k = \beta m_{k-1} + (1 - \beta) \nabla f_{S_k}(x_k),$$
$$x_{k+1} = x_k - \eta_k d_k, \qquad d_k = A_k^{-1} m_k,$$

where $A_k = G_k^{1/2}$, and $G_k = [\beta_2 G_{k-1} + (1 - \beta_2) \text{diag}(\nabla f_{S_k}(x_k) \nabla f_{S_k}(x_k)^T)] / (1 - \beta_2^k)$ [5, 19]. Unlike Vaswani et al. [19] that uses preconditioned gradient (which guarantees descent because of $\langle A_k^{-1} \nabla f_{S_k}(x_k), \nabla f_{S_k}(x_k) \rangle > 0$) to do line-search, we search along the preconditioned momentum direction, i.e. $d_k = A_k^{-1} m_k$ under the MSL framework (MSL-Adam). When encountering $\langle A_k^{-1} m_k, \nabla f_{S_k}(x_k) \rangle < 0$, we perform the restart for Adam by setting $m_k = \beta m_0 + (1 - \beta) \nabla f_{S_k}(x_k)$ and $G_k = [\beta_2 G_0 + (1 - \beta_2) \text{diag}(\nabla f_{S_k}(x_k) \nabla f_{S_k}(x_k)^T)] / (1 - \beta_2^k)$, in which $m_0 = 0$ and $G_0 = 0$. Thus, a descent direction is guaranteed and can be used in (3).

We first verify the effectiveness of our proposed momentum corrections and restart strategy in the deterministic setting for minimizing a 2-dimensional function $f(x, y)$ [20]. All approaches reduce the oscillation behavior of heavy ball momentum (using the optimal parameters adapted from Polyak [13]) in the convergence of variable y to its optimum (Figure 3(c)). Moreover, the decrease in the overall objective function of our algorithm is faster than other step sizes such as ALG-MAG proposed by Wang et al. [20] (Figure 3(a)). We emphasize that **for all our algorithms, namely MSL-SGDM-C, MSL-SGDM-R, and MSL-Adam, the momentum parameter β directly participates in the line-search through the generalized stochastic Armijo condition (i.e. (3)), which improves the stability of the algorithm for different values of β (Figure 1(*a*))**. In the next section, we present further experimental results in the stochastic setting.

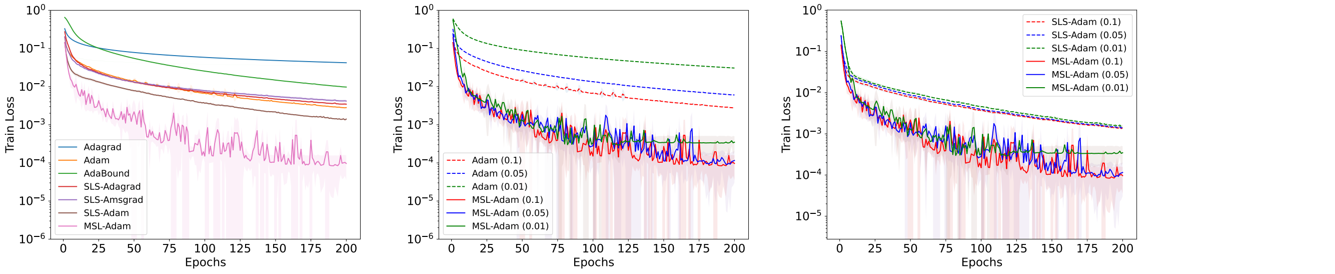


(a) Different algorithms (b) SGDM vs. MSL-SGDM-C (c) SGDM vs. MSL-SGDM-R

Figure 4: Results of binary classification on the synthetic dataset. (a) Train loss against epochs for different algorithms. (b) Compare the sensitivity of MSL-SGDM-C to search initialization $\tilde{\eta}_0$ and of SGDM to its step size. (c) Similar comparison as (b) but between MSL-SGDM-R and SGDM.

3. Experiments

We perform experiments on a binary classification task using a separable-synthetic dataset and logistic loss. We follow the protocol in Loizou et al. [8] to generate the dataset of margin τ. For the actual implementation of line-search, we use a back-tracking procedure where we start from $\tilde{\eta}_0$ and decrease it by a factor of γ every round until the condition (3) is met. To reduce the search cost, we adapt a strategy from Vaswani et al. [18] such that the search starting point at current iteration (denoted as $\tilde{\eta}_k$) is $\tilde{\eta}_k = \gamma^{b/n}\eta_{k-1}$, where b and n are the batch size and the total number of data points, respectively. For line-search experiments, we choose $\gamma = 2.0$, $b = 128$, and $c = 0.5$ in (3). For constant-step algorithms, we do a grid search on the step size in the values $[1.0, 0.1, 0.05, 0.01]$, and choose the one that minimizes the train loss. For the momentum parameter β, we set it to be 0.9 for all experiments in this section. For each experiment setting, we perform 5 parallel runs and compute the means and standard deviations. The standard deviations are shown as shaded regions in the graphs. We first compare MSL-SGDM-R and MSL-SGDM-C against other baselines, including constant step-size SGDM [7], SLS [18], ALG-SMAG [20], and SLS-SGDM. For SLS-SGDM, the search via (3) is based on $\nabla f_{S_k}(x_k)$, which does not involve the momentum parameter β. We highlight that our algorithm is faster than this approach as shown in Figure 4(a), which suggests the importance of using the actual update direction to perform line-search when momentum is present. Compared to constant step-size SGDM, which is highly sensitive to the choice of step size, either momentum corrections (MSL-SGDM-C) or the restart approach (MSL-SGDM-R) are robust to the search initialization, i.e. $\tilde{\eta}_0$, as demonstrated in Figure 4(b) and Figure 4(c). Finally, the extension of the restart framework to Adam, i.e. MSL-Adam, converges faster than the baselines Adagrad [2], Adam [5], AdaBound [10], SLS-Adagrad [19], SLS-Amsgrad [19], and SLS-Adam. We highlight that MSL-Adam uses the full Adam update direction (i.e. $d_k = A_k^{-1}m_k$) in line-search, whereas SLS-Adam only uses the preconditioned gradient. Similar to MSL-SGDM-C/R, MSL-Adam has fast convergence and is robust to different search initialization $\tilde{\eta}_0$s. We also notice that the decrease in training loss of our algorithms is less monotone compared to others. This is potentially caused by momentum corrections or restart making the update directions less correlated with each other.



(a) Different algorithms (b) Adam vs. MSL-Adam (c) SLS-Adam vs. MSL-Adam

Figure 5: Results of binary classification on the synthetic dataset. (a) Train loss against epochs for different algorithms. (b) Compare the sensitivity of Adam to its step size and of MSL-Adam to its search initialization $\tilde{\theta}_0$. (c) Similar comparison as (b) but between SLS-Adam and MSL-Adam.

4. Related Works

Besides stochastic Polyak step size (SPS) and stochastic line search (SLS) [8, 18], which have strong theoretical guarantees in the interpolating settings, some recent work has studied their extensions under non-interpolation. For example, Orvieto et al. [12] proposed a monotonically-decreasing variant of SPS for non-interpolating convex problems (DecSPS). Fan et al. [3] unified SLS and SPS under an Envelope-type step size (SLSB and SPSB), and relaxed the monotonicity requirement in the step size. Jiang and Stich [4] introduced Adagrad-type modifications to SPS (AdaSPS) and SLS (AdaSLS), and showed that the resulting step size can achieve optimal convergence rates in both interpolating and non-interpolating settings. When momentum is present, Wang et al. [20] proposed two step size variants that are based on heavy-ball momentum (ALG-HB) and moving average gradient (ALG-MAG), respectively, and extended them to the stochastic setting. Besides these adaptive step sizes for SGD or SGD with momentum, coordinate-wise adaptive step sizes such as Adam and its variants are popular for training large models such as transformers [5, 6, 9, 10]. Vaswani et al. [19] proposed using the preconditioned gradient line-search for Amsgrad and Adagrad. To the best of our knowledge, existing line-search methods in the stochastic setting rely on the search direction being a descent direction, and the incorporation of momentum into line-search is not fully explored.

5. Conclusion

In summary, we have utilized the generalized Armijo condition in the stochastic setting for momentum-based updates. For a non-descent direction, we propose using either momentum corrections or restart to fix the direction so that it guarantees descent. This leads to two new algorithms for SGD with momentum, namely MSL-SGDM-R and MSL-SGDM-C. Finally, we extend the restart approach to Adam which gives rises to the MSL-Adam algorithm. We empirically verify that our algorithms are robust to the choice of momentum parameter and other hyperparameters. For future directions, we are interested in analyzing the convergence rates of our proposed algorithms.

Acknowledgement This work is funded partially by the the Canada CIFAR AI Chair Program, NSERC Discovery Grant RGPIN-2021-03677, and the NSERC Discovery Grant RGPIN-2022-03669.

References

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

[2] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[3] Chen Fan, Gaspard Choné-Ducasse, Mark Schmidt, and Christos Thrampoulidis. Bisls/sps: Auto-tune step sizes for stable bi-level optimization. *arXiv preprint arXiv:2305.18666*, 2023.

[4] Xiaowen Jiang and Sebastian U Stich. Adaptive sgd with polyak stepsize and line-search: Robust convergence and variance reduction. *arXiv preprint arXiv:2308.06058*, 2023.

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

[6] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

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

[8] Nicolas Loizou, Sharan Vaswani, Issam Hadj Laradji, and Simon Lacoste-Julien. Stochastic polyak step-size for sgd: An adaptive learning rate for fast convergence. In *International Conference on Artificial Intelligence and Statistics*, pages 1306–1314. PMLR, 2021.

[9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[10] Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. *arXiv preprint arXiv:1902.09843*, 2019.

[11] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[12] Antonio Orvieto, Simon Lacoste-Julien, and Nicolas Loizou. Dynamics of sgd with stochastic polyak stepsizes: Truly adaptive variants and convergence to exact solution. *Advances in Neural Information Processing Systems*, 35:26943–26954, 2022.

[13] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

[14] Michael James David Powell. Restart procedures for the conjugate gradient method. *Mathematical programming*, 12:241–254, 1977.

[15] Othmane Sebbouh, Robert M Gower, and Aaron Defazio. Almost sure convergence rates for stochastic gradient descent and stochastic heavy ball. In *Conference on Learning Theory*, pages 3935–3971. PMLR, 2021.

[16] Jonathan Richard Shewchuk et al. An introduction to the conjugate gradient method without the agonizing pain, 1994.

[17] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013.

[18] Sharan Vaswani, Aaron Mishkin, Issam Laradji, Mark Schmidt, Gauthier Gidel, and Simon Lacoste-Julien. Painless stochastic gradient: Interpolation, line-search, and convergence rates. *Advances in neural information processing systems*, 32, 2019.

[19] Sharan Vaswani, Issam Laradji, Frederik Kunstner, Si Yi Meng, Mark Schmidt, and Simon Lacoste-Julien. Adaptive gradient methods converge faster with over-parameterization (but you should do a line-search). *arXiv preprint arXiv:2006.06835*, 2020.

[20] Xiaoyu Wang, Mikael Johansson, and Tong Zhang. Generalized polyak step size for first order optimization with momentum. *arXiv preprint arXiv:2305.12939*, 2023.