Generalized Polyak Step Size for First Order Optimization with Momentum

Xiaoyu Wang [1] **Mikael Johansson** [2] **Tong Zhang** [1]

Abstract

In machine learning applications, it is well known that carefully designed learning rate (step size) schedules can significantly improve the convergence of commonly used first-order optimization algorithms. Therefore how to set step size adaptively becomes an important research question. A popular and effective method is the Polyak step size, which sets step size adaptively for gradient descent or stochastic gradient descent without the need to estimate the smoothness parameter of the objective function. However, there has not been a principled way to generalize the Polyak step size for algorithms with momentum accelerations. This paper presents a general framework to set the learning rate adaptively for first-order optimization methods with momentum, motivated by the derivation of Polyak step size. It is shown that the resulting techniques are much less sensitive to the choice of momentum parameter and may avoid the oscillation of the heavy-ball method on ill-conditioned problems. These adaptive step sizes are further extended to the stochastic settings, which are attractive choices for stochastic gradient descent with momentum. Our methods are demonstrated to be more effective for stochastic gradient methods than prior adaptive step size algorithms in large-scale machine learning tasks.

1. Introduction

We consider stochastic optimization problems on the form

$$\min_{x \in \mathbb{R}^d} f(x) := \mathbb{E}_{\xi \sim \Xi}[f(x;\xi)] \qquad (1)$$

where ξ is a random variable with probability distribution Ξ and $f(x;\xi)$ is the instantaneous realization of f with respect to ξ. We use X^* to denote the set of minimizers of (1), which we assume is non-empty. In other words, there is at least one $x^* \in \mathbb{R}^d$ such that $f^* = f(x^*) = \min f(x)$.

Stochastic gradient descent (SGD) (Robbins & Monro, 1951) has been the workhorse for training machine learning models. To accelerate its practical performance, one often adds a momentum term to SGD, leading to algorithms such as SGDM (Sutskever et al., 2013). SGDM has been widely used in deep neural networks due to its empirical success and is a default choice in machine learning libraries (PyTorch and TensorFlow). However, its practical performance relies heavily on the choice of the step size (learning rate) that controls the rate at which the model learns.

In the traditional optimization literature, Polyak's heavy ball (momentum) method (Polyak, 1964) is a well-known technique to accelerate gradient descent. By accounting for the history of the iterates, it achieves a linear convergence that is substantially faster than gradient descent on ill-conditioned problems. However, to achieve its optimal performance, the heavy-ball method relies on a specific combination of momentum parameter β and step size η adapted to the condition number of the problem. One downside of the method is that its empirical performance is very sensitive to the momentum factor $\beta \in (0,1)$ (see Figure 4), which makes the method difficult to use when the condition number is unknown. For the heavy-ball method, each component of the decision vector is updated independently and shares the same step size. Even in very simple examples, the method exhibits a zigzag phenomenon in the dimension with large curvature (Polyak, 1964), leading to a slow and oscillatory convergence. Thus, an adaptive step size is important for the best practical behavior of the heavy-ball method.

For convex functions whose optimal value is known a priori, the Polyak step size, which depends on the current function value and the magnitude of the gradient (subgradient), is optimal in a certain sense (Polyak, 1987; Camerini et al., 1975; Brännlund, 1995). Hazan & Kakade (2019) revisited the Polyak step size and proved near-optimal convergence even when the optimal value is unknown. Still, it is impractical to use the deterministic Polyak step size due to the computation of exact function values and gradients in each iteration. Recently, there has been a strong interest in developing adaptive step size policies that are inspired by the classical Polyak step size (Rolinek & Martius, 2018;

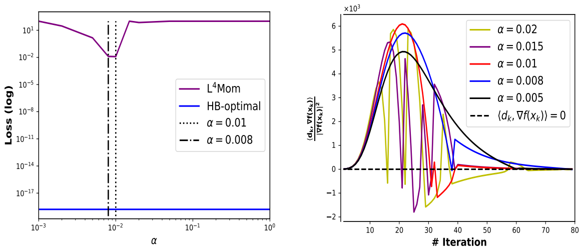
[1] The Hong Kong University of Science and Technology [2] Royal Institute of Technology (KTH). Correspondence to: Xiaoyu Wang <maxywang@ust.hk>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

Prazeres & Oberman, 2021; Berrada et al., 2020; Loizou et al., 2021; Sebbouh et al., 2021). This line of research has been particularly successful on overparameterized models. Loizou et al. (2021) extended the Polyak step size to the stochastic setting and proposed a stochastic Polyak step size (SPS). Berrada et al. (2020) made explicit use of the interpolation property to design a step size policy for SGD in a closed form (called ALI-G) and incorporated regularization as a constraint to promote generalization. The experiments in Berrada et al. (2020; 2021) used momentum without and theoretical guarantee and demonstrated that it could significantly improve the practical performance. This highlights the importance of adaptive step size policies for momentum methods. However, the existing research on Polyak step sizes has focused on SGD, and rarely designed adaptive step sizes for heavy-ball and momentum algorithms.

1.1. Motivation

To demonstrate the challenges that arise in adapting the Polyak step size to momentum algorithms, we consider the approach that underpins L^{4}Mom (Rolinek & Martius, 2018). The key idea is to linearize the loss function at the current iterate, $f(x_k - \eta d_k) \approx f(x_k) - \eta \langle \nabla f(x_k), d_k \rangle$ and then choose η_k so that the linearized prediction of f at the next iterate equals $f^\star$. To account for the inaccuracy of the linear approximation, L^{4}Mom introduces a hyperparameter $\alpha > 0$ and uses $\eta = \alpha \frac{f(x_k) - f^*}{\langle \nabla f(x_k), d_k \rangle}$. However, we have found that this algorithm is quite unstable in practice, and fails on standard experiments such as the CIFAR100 experiments in Section 5.2. This sensitivity is also observed in (Berrada et al., 2020). One reason is that $\langle \nabla f(x_k), d_k \rangle$ is not always guaranteed to be positive. We experience difficulties with the algorithm even in a simple least-squares problem with condition number $\kappa = 10^4$ and $f^* = 0$. The parameter α is crucial for the empirical convergence: for large values of α, the algorithm easily explodes, while small values of α result in slow convergence. In brief, L^4 is not an ideal approach for finding an adaptive step size for heavy-ball or momentum acceleration. Besides, there is no theoretical guarantee for the L^{4}Mom algorithm. Our goal is to find adaptive step sizes for the momentum acceleration algorithms that are more stable and efficient in practice.



1.2. Contribution

A new perspective on adaptive step sizes for momentum. Inspired by the success of the Polyak step size for subgradient methods and SGD, and the absence and insufficiency of adaptive Polyak step sizes for momentum accelerations, we propose a generic Adaptive Learning Rate (ALR) framework for two variants of momentum methods: heavy-ball (HB) and moving averaged gradient (MAG). We call corresponding adaptive algorithms ALR-HB and ALR-MAG and make the following contributions:

(i) We prove global linear convergence of ALR-MAG on semi-strongly convex and smooth functions, improving the results for modified subgradient methods in (Brännlund, 1995), under less restrictive assumptions.

(ii) For least-squares problems, we demonstrate that ALR-HB and ALR-MAG are less sensitive to the choice of β than the original heavy-ball method. Our algorithms are significantly better than heavy-ball, gradient descent with Polyak step size, and L^{4}Mom if the condition number is unknown a priori.

(iii) The proposed framework is also applicable to Nesterov accelerated gradient (NAG) (Nesterov, 1983) and performs better than the original Nesterov momentum under optimal parameters (see Appendix A).

Stochastic extensions of ALR-HB and ALR-MAG. We extend ALR-HB and ALR-MAG to the stochastic setting and call them ALR-SHB and ALR-SMAG, respectively. We make the following contributions:

(i) Under the assumption of interpolation (overparameterized models), we prove a linear convergence rate for ALR-SMAG on semi-strongly convex and smooth functions. Such a result did not exist for SGD with momentum under this class of step sizes.

(ii) We demonstrate the superiority of ALR-SHB and ALR-SMAG over state-of-the-art adaptive methods and the popular step-decay step size (Ge et al., 2019) on logistic regression and deep neural network training. By incorporating a warmup technique into the upper bound of the step size, the performance of ALR-SHB and ALR-SMAG can be improved further and performs better than step-decay.

(iii) We incorporate weight-decay into the update of ALR-SMAG to improve the generalization. The algorithm performs better than ALI-G with momentum and step-decay step size and is comparable to cosine step size without restart (Loshchilov & Hutter, 2017).

2. Adaptive Step Sizes

Consider a general first-order method with momentum acceleration on the form

$$x_{k+1} = x_k - \eta_k d_k + \gamma(x_k - x_{k-1}) \tag{2}$$

where $-d_k$ is a descent direction. A natural question that arises is how far we should move in this direction to converge quickly. In theoretical analyses, the quantity $\|x - x^*\|^2$ is often used to measure the convergence of the algorithms. We therefore propose to optimize η_k to ensure that $x_{k+1}(\eta_k)$ minimizes this quantity, *i.e.*,

$$\min_{\eta_k} \|x_{k+1}(\eta_k) - x^*\|^2 . \tag{3}$$

Minimizing (3) w.r.t η_k suggests that

$$\eta_k = \frac{\langle d_k, x_k - x^* \rangle}{\|d_k\|^2} + \gamma \frac{\langle d_k, x_k - x_{k-1} \rangle}{\|d_k\|^2}. \tag{4}$$

In general, the minimizer x^* is not accessible. However, when f is convex, we can often evaluate a lower bound of $\langle d_k, x_k - x^* \rangle$ and minimize an upper bound of (3)

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &= \|x_k - x^\star\|_2^2 + \eta_k^2 \|d_k\|_2^2 \\ &\quad - \eta_k \gamma \langle d_k, x_k - x_{k-1} \rangle - \eta_k \langle d_k, x_k - x^* \rangle . \end{aligned} \tag{5}$$

For example, if $d_k \in \partial f(x_k)$ and $\gamma = 0$, the method of (2) reduces to the subgradient method. By the convexity of f, $\langle \nabla f(x_k), x_k - x^* \rangle \geq f(x_k) - f^*$, and minimizing the upper bound of (5) results in the Polyak step size $\eta_k = (f(x_k) - f^*)/\|d_k\|^2$ (Bazaraa & Sherali, 1981). Whatever other model we may have that provides a lower bound on the inner product $\langle d_k, x_k - x^* \rangle$ will also work in this framework. In the rest of this paper, we focus on two popular variants of momentum acceleration.

2.1. Adaption for Heavy-Ball

We first consider the heavy-ball (HB) method (Polyak, 1964; Ghadimi et al., 2015) given by

$$x_{k+1} = x_k - \eta_k \nabla f(x_k) + \beta(x_k - x_{k-1}) \tag{6}$$

where $\beta \in (0, 1)$ is a constant. Clearly, heavy ball is a special case of (2) where $\gamma = \beta$ and $d_k = \nabla f(x_k)$. By the convexity of f, we have a lower bound for $\langle \nabla f(x_k), x_k - x^* \rangle$ by $f(x_k) - f^*$ and minimizing the upper bound of (5) yields the adaptive learning rate for heavy-ball (*ALR-HB*)

$$\eta_k = \frac{f(x_k) - f^*}{\|\nabla f(x_k)\|^2} + \beta \frac{\langle \nabla f(x_k), x_k - x_{k-1} \rangle}{\|\nabla f(x_k)\|^2}. \tag{7}$$

If the objective function f is also L-smooth then $\langle \nabla f(x_k), x_k - x^* \rangle \geq f(x_k) - f(x^*) + \frac{1}{2L} \|\nabla f(x_k)\|^2$, which is a tighter lower bound for $\langle \nabla f(x_k), x_k - x^* \rangle$. It results in the formula (8) below, named ALR-HB(v2), which has an additional constant term $1/(2L)$ compared to (7):

$$\eta_k = \frac{1}{2L} + \frac{f(x_k) - f^*}{\|\nabla f(x_k)\|^2} + \beta \frac{\langle \nabla f(x_k), x_k - x_{k-1} \rangle}{\|\nabla f(x_k)\|^2}. \tag{8}$$

Algorithm 1 ALR-HB

```
Input: initial point $x_1$, $\beta \in (0, 1)$, $v_0 = \mathbf{0}$
while $x_k$ does not converge do do
    $k \leftarrow k + 1$
    $\eta_k \leftarrow \frac{f(x_k) - f(x^*)}{\|\nabla f(x_k)\|^2} + \beta \frac{\langle \nabla f(x_k), x_k - x_{k-1} \rangle}{\|\nabla f(x_k)\|^2}$ (v1);
    $\eta_k \leftarrow \frac{1}{2L} + \frac{f(x_k) - f^*}{\|\nabla f(x_k)\|^2} + \beta \frac{\langle \nabla f(x_k), x_k - x_{k-1} \rangle}{\|\nabla f(x_k)\|^2}$ (v2)
    $v_k \leftarrow -\eta_k \nabla f(x_k) + \beta v_{k-1}$
    $x_{k+1} \leftarrow x_k + v_k$
end while
```

The ALR-HB algorithms are shown in Algorithm 1. Our next example shows that ALR-HB (v2) can find the exact solution for a simple least-squares problem in a single step.

Example 1. *Consider one-dimensional least-squares problem $f(x) = \frac{1}{2} h x^2$. For Polyak with gradient descent and L^4Mom, we need at least $k = \log_2(x_0/\epsilon)$ steps for an ϵ-accurate solution ($|x - x^*| \leq \epsilon$). For ALR-HB(v2), given x_0, x_1, we only need one step to find the exact solution.*

Proof. The step size of ALR-HB(v2) can be written as

$$\begin{aligned} \eta_k &= \frac{1}{2L} + \frac{f(x_k) - f^*}{\|\nabla f(x_k)\|^2} + \beta \frac{\langle \nabla f(x_k), x_k - x_{k-1} \rangle}{\|\nabla f(x_k)\|^2} \\ &= \frac{1}{h} + \beta \frac{1}{h} \left(1 - \frac{x_{k-1}}{x_k} \right) \end{aligned}$$

Applying the step size to the iterate of HB gives

$$x_{k+1} = x_k - \eta_k h x_k + \beta(x_k - x_{k-1}) = 0.$$

□

Thus, we believe that the model (3) is a good choice for designing adaptive step sizes for the heavy-ball method.

2.2. Adaptive Step Size for MAG

Next, we consider the moving averaged gradient (MAG), another widely used momentum variant for deep learning

$$d_k = \nabla f(x_k) + \beta d_{k-1}, \quad x_{k+1} = x_k - \eta_k d_k \tag{9}$$

where $\beta \in (0, 1)$. Note that if the step size is constant, $\eta_k = \eta$, then the formulas (6) and (9) are equivalent. However, we consider adaptive step sizes that change with k, and in this case, the two methods are different variants of momentum.

If the search direction d_k is defined by (9) and $\gamma = 0$, the update of (2) reduces to the MAG algorithm. By the convexity of f, if $\eta_i \leq \frac{f(x_i)-f^*}{\|d_i\|^2}$ for all $i \leq k-1$, Lemma 4.2 in our subsequent theoretical analysis shows that $\langle d_{k-1}, x_k - x^* \rangle \geq 0$. We therefore provide a lower bound for $\langle d_k, x_k - x^* \rangle = \langle \nabla f(x_k) + \beta d_{k-1}, x_k - x^* \rangle \geq \langle \nabla f(x_k), x_k - x^* \rangle \geq f(x_k) - f^*$. Minimizing the upper bound of (5) results in step size:

$$\eta_k = \frac{f(x_k) - f^*}{\|d_k\|^2}. \tag{10}$$

We refer to this adaptive momentum version, detailed in Algorithm 2, as ALR-MAG. Lemma 4.3 in Section 4.1 shows that the iterates of ALR-MAG decrease monotonically w.r.t. the distance $\|x - x^*\|^2$. This guarantees that the iterates come closer and closer to the optimum. Our next example in Section 2.3 demonstrates that the step size of ALR-MAG is able to avoid oscillations of the heavy-ball method.

Algorithm 2 ALR-MAG

```
Input: initial point $x_1$, $\beta \in (0, 1)$, $d_0 = \mathbf{0}$
while $x_k$ does not converge do
    $k \leftarrow k + 1$
    $d_k \leftarrow \beta d_{k-1} + \nabla f(x_k)$
    $\eta_k \leftarrow \frac{f(x_k)-f^*}{\|d_k\|^2}$
    $x_{k+1} \leftarrow x_k - \eta_k d_k$
end while
```

2.3. Justification of ALR-MAG

To demonstrate the advantages of ALR-MAG, we consider a simple two-dimensional least-squares problem $f(x, y) = \frac{1}{2}(x-1)^2 + \frac{\kappa}{2}(y+1)^2$ with $x \in \mathbb{R}$ and $y \in \mathbb{R}$. We set $\kappa = 100$ and use the initial point $(x_0, y_0) = (48, -28)$. For the classic heavy-ball method, the iterates can be re-written as $x_{k+1} = x_k - \eta_k(x_k - 1) + \beta(x_k - x_{k-1}); y_{k+1} = y_k - \eta_k\kappa(y_k + 1) + \beta(y_k - y_{k-1})$. Note that the variables x and y are updated independently and share the same step size. Our baseline is the optimal parameters for heavy-ball from (Polyak, 1964), $\beta^* = (\sqrt{\kappa}-1)^2/(\sqrt{\kappa}+1)^2$ and $\eta^* = (1+\sqrt{\beta^*})^2/L$ (called HB-optimal). From Figure 1(left), we observe a pronounced zigzag behavior in the y-dimension for HB-optimal. The step size η^* is large and results in an undamped and slow convergence in the dimension with large curvature (*i.e.,* y). We apply ALR-MAG with the same β^* and $f^\star = 0$. ALR-MAG adapts the step size to start from a small value to avoid the instability in the y-dimension and finally reaches a value that is comparable to η^*.

3. Related Work

Adaptive methods for deterministic momentum. For deterministic problems with μ-strongly convex and L-smooth objective functions, Polyak (1964) demonstrated that the fastest local convergence of the heavy-ball method is attained for the optimal parameters $\beta^* = (\sqrt{\kappa}+1)^2/(\sqrt{\kappa}+1)^2$ and $\eta^* = (1+\sqrt{\beta^*})^2/L$ where $\kappa = L/\mu$ is the condition number. Fast linear convergence is also achieved for Nesterov's accelerated gradient method with $\beta = (\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)$ and $\eta = 1/L$ (Nesterov, 2003). However, while L is relatively easy to estimate on-line, μ (and therefore κ) is often inaccessible. A number of recent contributions suggest ideas for approximating these optimal hyper-parameters at each iteration. Barré et al. (2020) adaptively estimate the strong convexity constant by the inverse of the Polyak step size and use this estimate in place of the true μ in the momentum parameter for Nesterov momentum. Saab et al. (2022) approximate the Lipschitz and strongly convexity constants employing the absolute differences of current and previous model parameters and their gradients. However, its empirical performance, at least in the least-squares problem in Figure 2 (labeled *AHB*), is poor. Malitsky & Mishchenko (2020) estimate the Lipschitz constant similarly to (Saab et al., 2022) and the strong convexity constant μ by the inverse smoothness of the conjugate function. They also add a conservation bound for the estimators of μ, L, leading to a method with four hyperparameters that need to be tuned.

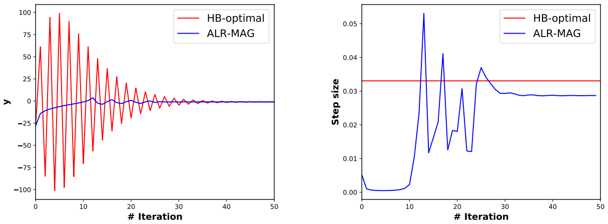


Figure 1. The trace of variable y (left) and the step sizes (right)

Adaptive step sizes for stochastic algorithms. Vaswani et al. (2019) extend line search methods to the stochastic setting (called SLS) using the function and gradient of a mini-batch and guarantee linear convergence under interpolation. But its many hyper-parameter makes it difficult to use in practice. Malitsky & Mishchenko (2020) use their estimation technique for L (discussed above) to develop an adaptive step size for SGD (called *AdSGD*). Under interpolation, the iteration complexity is κ times higher than SGD.

Adaptive gradient methods. Adaptive gradient methods, such as AdaGrad (Duchi et al., 2011), RMSProp (Tieleman & Hinton, 2012), Adam (Kingma & Ba, 2015), and AdamW (Loshchilov & Hutter, 2018) are very popular in practice. However, adaptive gradient methods have poor generalization compared to SGD in supervising learning tasks (Wilson et al., 2017). Liu et al. (2019) suggest a learning rate warmup heuristic in the early stage of training that can improve the generalization of adaptive methods.

4. Preliminaries and Convergence Analysis

Before presenting our theoretical results, we introduce a few key concepts and the notation used throughout the paper.

Definition 4.1. $f : \mathbb{R}^d \mapsto \mathbb{R}$ is semi-strongly convex if there exists a constant $\hat{\mu} > 0$ such that $\frac{\hat{\mu}}{2} \|x - x^*\|^2 \leq f(x) - f^*, \forall x \in \mathbb{R}^d$.

This condition is also called the quadratic growth property of f. If the function is convex and smooth, semi-strong convexity is equivalent to the Polyak-Łojasiewicz (PL) condition (Karimi et al., 2016). This is a weaker condition than the strong convexity. The definitions of convexity, strong convexity, and L-smoothness are provided in Appendix B.

Interpolation. We say that the interpolation condition holds if there exists $x^* \in \mathcal{X}^*$ such that individual functions $\min_x f(x;\xi) = f(x^*;\xi)$ for all $\xi \in \Xi$. All loss functions $f(x;\xi)$ meet with a common minimizer x^*. The interpolation property is satisfied in many machine learning models, including linear classifiers with separable data, over-parameterized deep neural networks (Ma et al., 2018; Zhang et al., 2021), non-parametric regression (Liang & Rakhlin, 2020), and boosting (Bartlett et al., 1998).

4.1. ALR-MAG in Deterministic Optimization

We first provide convergence guarantees for the ALR-MAG method on deterministic convex optimization problems where the exact gradient and function values are available.

Our first lemma shows that the direction d_{k-1} forms an acute angle with the direction from x_k to the minimizer x^*.

Lemma 4.2. *Let f be convex and assume that $\{x_i\}_{i=0}^k$ has been generated by ALR-MAG with $\eta_i \leq (f(x_i) - f^*)/\|d_i\|^2$ for all $i \leq k-1$. Then, $\langle d_{k-1}, x_k - x^* \rangle \geq 0$.*

The next lemma establishes that the MAG iterates under the step size (10) are monotone decreasing with respect to the distance $\|x - x^*\|^2$. This guarantees that the next iterate x_{k+1} is closer to the minimizer x^* than the current x_k.

Lemma 4.3. *Let $\{x_k\}$ be generated by MAG with the step size defined in (10). Then, if f is convex*

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \eta_k \left(f(x_k) - f^*\right).$$

If, in addition, f is L-smooth, then

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \left(\eta_k + \frac{(1-\beta)}{L}\right)(f(x_k) - f^*).$$

From Lemma 4.3, we can note that if f is smooth, there is an extra decrease compared to when f is only convex. If the function is also semi-strongly convex, then we have the following linear convergence result.

Theorem 4.4. *Suppose that function f is convex and L-smooth and consider the ALR-MAG algorithm under the step size (10). If f is semi-strongly convex with $\hat{\mu} > 0$, then*

$$\|x_k - x^*\|^2 \leq (1-\rho)^k \|x_1 - x^*\|^2 \tag{11}$$

where $\rho = (1-\beta)(2\kappa)^{-1}$ and $\kappa = L/\hat{\mu}$.

Brännlund (1995) generalizes the subgradient method to use a convex combination of previous subgradients. Hence, the MAG algorithm (9) can be seen as a special case of Brännlund (1995). Since the sequence $\|x_{k+1} - x^*\|^2$ is monotone decreasing, it is enough to require L-smoothness for all x with $\|x - x^\star\| \leq \|x_0 - x^\star\|$, which matches the smoothness assumption of Theorem 2.6 in (Brännlund, 1995). However, compared to (Brännlund, 1995), we do not have the extra restriction $\|d_k\| \leq \|\nabla f(x_k)\|$. In fact, this requirement on d_k is, in general, not satisfied for $\beta \in (0,1)$. Note that Theorem 4.4 improves the dependence of the condition number κ from κ^2 to κ in the convergence of (Brännlund, 1995; Shor, 2012) and only requires semi-strong convexity (while Theorem 2.12 in (Shor, 2012) assumes strong convexity). More results of ALR-MAG for functions without smoothness are provided in Appendix B.

4.2. Convergence of ALR-MAG in Stochastic Settings

Next, we extend the adaptive step size for HB and MAG to the case when gradients and function values are sampled from an underlying (data) distribution. It is typically not practical to compute the exact function value and gradient in every step. Instead, we evaluate a mini-batch S_k of gradient and function value samples in each iteration

$$f_{S_k}(x) = \frac{1}{|S_k|} \sum_{i \in S_k} f(x;\xi_i), \nabla f_{S_k}(x) = \frac{1}{|S_k|} \sum_{i \in S_k} \nabla f(x;\xi_i)$$

and propose to use the following adaption of (10):

$$\eta_k = \min\left\{\frac{f_{S_k}(x_k) - f_{S_k}^*}{c\|d_k\|^2}, \eta_{\max}\right\} \tag{12}$$

Here, $d_k = \beta d_{k-1} + \nabla f_{S_k}(x_k)$ and $f_{S_k}^* = \inf_x f_{S_k}(x)$. We refer to the stochastic version of MAG as *SMAG*, and as *ALR-SMAG* when we use the adaptive step size (12).

Their step size (12) has three modifications compared to (10). First, while the immediate extension of (10) would replace $f^* = f(x^*)$ by $f_{S_k}(x^*)$, we suggest to use $f_{S_k}^*$ instead. For example, in many machine learning problems with unregularized surrogate loss functions, we have $f_{S_k}^* = 0$ (Bartlett et al., 2006). For the loss with regularization for example ℓ_2 regularization, when the mini-batches contain a single data, then $f_{S_k}^*$ can be computed in a closed form for some standard loss function (Loizou et al., 2021; Bartlett et al., 2006). Second, we introduce a hyper-parameter $c > 0$ that controls the scale of the step size to account for the inaccuracy in function and gradients. Third, due to the convergence reason and to make it applicable to wide applications even nonconvex problems, we may restrict the step size to be upper bounded by $\eta_{\max} > 0$.

In the following results, we assume the finite optimal objective function difference which has been used in the analysis of stochastic Polyak step size (Loizou et al., 2021).

Assumption 4.5. (**Finite optimal objective difference**)

$$\sigma^2 = \mathbb{E}[f_{S_k}(x^*) - f_{S_k}^*] = f(x^*) - \mathbb{E}[f_{S_k}^*] < +\infty$$

where $f_{S_k}^* = \inf f_{S_k}(x)$.

Under interpolation, each individual function $f(x;\xi)$ attains its optimum at x^* which implies that $\sigma = 0$. We focus on semi-strongly convex and smooth functions.

Theorem 4.6. *Suppose that the individual function $f(x;\xi)$ is convex and L-smooth for any $\xi \in \Xi$ and that Assumption 4.5 holds. Consider ALR-SMAG with $c > 1$, if f is semi-strongly convex with $\hat{\mu}$, then*

$$\mathbb{E}[\|x_{K+1} - x^*\|^2] \leq (1-\rho_1)^K \|x_1 - x^*\|^2 + \frac{2\eta_{\max}\sigma^2}{\rho_1(1-\beta)}$$

where $\rho_1 = \min\left\{\frac{(1-\beta)(c-1)\hat{\mu}}{2c^2L}, \frac{(2c-1)\hat{\mu}\eta_{\max}}{2c}\right\}$.

When $\beta = 0$, step size (12) reduces to SPS_max. Our result in Theorem 4.6 is comparable to theorem 3.1 of SPS_max for strongly convex functions. However, the numerical results show the superior performance of ALR-SMAG compared to SPS in a wide range of machine learning applications. In Theorem 4.6, we assume $c > 1$ to ensure that the step size is not too aggressive. For example, in the experiments on logistic regression in Section C.2, we will use $c = 5$. For the deep learning tasks (nonconvex), we suggest that $c < 1$. This coincides with parameter c from SPS_max (they set $c = 0.2$) (Loizou et al., 2021).

An important property of SPS (Loizou et al., 2021) is that the step size is lower and upper-bounded. This is not the case for our step size. Since d_k is a convex combination of all previous stochastic gradients, the scale of d_k is controlled by the previous stochastic gradients. In general, it is not clear how $\|d_k\|$ is related to $\|\nabla f_{S_k}(x_k)\|$, which makes it challenging to analyze the convergence of SMAG under (12). A key step in our analysis is to establish the inequality (18) to handle the moving averaged gradient. The main novelty of ALR-SMAG is that it provides a principled way to adapt the step size for SGD with momentum and guarantees linear convergence, which earlier techniques were unable to do (Rolinek & Martius, 2018; Berrada et al., 2020; 2021).

The constant term in the inequality in Theorem 4.6 can not be made arbitrarily small by decreasing the upper bound $\eta_{\max}$. We also observe this limitation in the stochastic Polyak step size; see Theorem 3.1 and Corollary 3.3 in SPS (Loizou et al., 2021). Theorem 4.6 suggests that $\beta = 0$ achieves the best result in theory. This is also an issue for the stochastic momentum analysis (Yan et al., 2018; Liu et al., 2020).

Our next corollary provides a stronger convergence result if the model is expressive enough to interpolate the data. In this setting, we use no maximal learning rate.

Corollary 4.7. *Assume interpolation ($\sigma = 0$) and suppose that all assumptions of Theorem 4.6 hold. Consider the step size (12) and $\eta_{\max} = \infty$. Then*

$$\mathbb{E}[\|x_{K+1} - x^*\|^2] \leq \left(1 - \rho_1^{'}\right)^K \|x_1 - x^*\|^2$$

where $\rho_1^{'} = \frac{(1-\beta)(c-1)\hat{\mu}}{2c^2L}$.

Under interpolation, ALR-SMAG can converge to the optimal solution x^* and achieves the fast linear convergence rate $\mathcal{O}\left((1-(1-\beta)\hat{\mu}/L)^k\right)$ under semi-strong convexity. We also provide the convergence results of ALR-SMAG for general convex functions in Theorem B.10 (see Appendix B).

In the end, we will compare the analysis above with other adaptive step sizes and stochastic momentum methods. The iterate complexity of AdSGD (Malitsky & Mishchenko, 2020) is κ higher compared to SGD for adaptive estimation of the stepsize. Clearly, the complexity of ALR-SMAG under interpolation is better than that of AdSGD. SMAG under constant step size is equivalent to stochastic heavy-ball (SHB) (Yan et al., 2018) and SGDM (Liu et al., 2020). In proposition 2 (Liu et al., 2020), the constant step size is restricted to be smaller than a small number $(1-\beta)/(5L)$ when the common choice $\beta = 0.9$ is applied. While Corollary 4.7 does not have any restriction for $\eta_{\max}$.

4.3. Stochastic Extension of ALR-HB.

The same idea to ALR-SMAG in Section 4.2, we consider applying the mini-batch of the function $f_{S_k} = \frac{1}{|S_k|}\sum_{i\in S_k} f(x;\xi_i)$ to the framework (3). A natural extension of ALR-HB to the stochastic setting is

$$\eta_k = \min\left\{\frac{f_{S_k}(x_k) - f_{S_k}^*}{c\|\nabla f_{S_k}(x_k)\|^2} + \frac{\beta\langle\nabla f_{S_k}(x_k), x_k - x_{k-1}\rangle}{\|\nabla f_{S_k}(x_k)\|^2}, \eta_{\max}\right\}. \tag{13}$$

We call the stochastic version of HB as SHB, and the algorithm SHB with step size (13) as ALR-SHB. Three changes are made compared to the direct generalization of (7). A similar discussion can be found in Section 4.2, which we omit here. In Appendix B.3, we provide a theoretical guarantee for truncated ALR-HB on least-squares but leave other possible results of ALR-HB and ALR-SHB for the future.

5. Numerical Evaluations

In this section, we evaluate the practical performance of the proposed adaptive step sizes. We start with experiments on the least-squares problems for ALR-MAG and ALR-HB, and continue by exploring the performance of the stochastic versions, ALR-SMAG and ALR-SHB, on large-scale convex optimization problems and deep neural networks training. For space concerns, the experiments in the convex interpolation setting are reported in appendix C.2.

5.1. Empirical Results on Ill-Conditioned Least-Squares

We use the procedure described in (Lenard & Minkoff, 1984) to generate test problems with $f(x) = \frac{1}{2}\left\|Ax - b\right\|^2$ where $A \in \mathbb{R}^{d_1 \times d}$ is positive definite, $b \in \mathbb{R}^{d_1}$ is a random vector, and the optimum $f^* = 0$. We report results for problems with $d_1 = d = 1000$ for which the condition number κ of $A^T A$ is 10^4. The strong convexity constant μ and smoothness constant L are the smallest and largest eigenvalues of the matrix $A^T A$, respectively.

We test *ALR-HB* and *ALR-MAG* against with several important methods including (1) gradient descent with Polyak step size (named GD-Polyak); (2) heavy-ball with the optimal parameters, i.e., $\beta^* = (\sqrt{\kappa} - 1)^2/(\sqrt{\kappa} + 1)^2$ and step size $\eta^* = (1 + \sqrt{\beta^*})^2/L$ (Polyak, 1964) (named HB-optimal); (3) L^4Mom (Rolinek & Martius, 2018); (4) AGM (variant II) (Barré et al., 2020); (5) AHB (Saab et al., 2022); (6)AdGD-accel (Malitsky & Mishchenko, 2020).

If μ and L are known a priori, we set $\beta = \beta^*$ for ALR-HB and ALR-MAG, as HB-optimal. We perform a grid search for parameters that are not specified (see Appendix C.1). The results are shown in Figure 2. ALR-HB (v2) performs the best among these methods. For the case that μ and L are not known, momentum parameter β is tuned from $\{0.5, 0.9, 0.95, 0.99\}$ for HB, L^4Mom, ALR-HB, and ALR-MAG. The results in Figure 3 show that ALR-HB and ALR-MAG are much better than HB with best-tuned constant step size, L^4Mom, and the other algorithms.

To illustrate the behavior of different step sizes, we plot ALR-HB and ALR-MAG with the theoretically optimal step size η^* for HB in Figure 2(right). We can see how ALR-HB (v2) oscillates in a small range around η^* and ALR-MAG converges to a value that is slightly different than η^*. Without knowledge of μ and L, ALR-HB still varies around η^* and captures the function's curvature (see Figure 3(right)). We also plot the final loss of the algorithms in Figure 4 on different β selected from the interval $[0.9, 1)$, which includes β^*. Clearly, ALR-HB is less sensitive to β than the original heavy-ball method while L^4Mom is far worse.

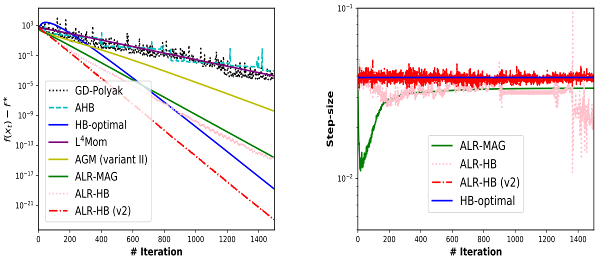


Figure 2. Least-squares with knowledge of μ and L (left: sub-optimality; right: step size)

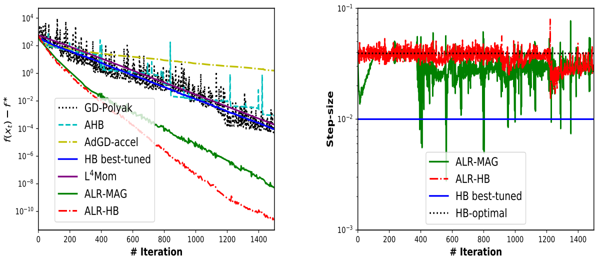


Figure 3. Least-squares without knowledge of μ and L (left: sub-optimality; right: step size)

5.2. Experimental Results on Deep Neural Networks

To show the practical implications of ALR-SMAG and ALR-SHB, we conduct experiments with deep neural network training on the CIFAR (Krizhevsky et al., 2009) and Tiny-ImageNet200 (Le & Yang, 2015) datasets. We compare ALR-SMAG and ALR-SHB against SGD with momentum (SGDM) under: constant step sizes; step-decay (Ge et al., 2019; Wang et al., 2021), where the step size is divided by 10 after the same number of iterations, and cosine decay without restart (Loshchilov & Hutter, 2017); the adaptive step size methods SPS_max (Loizou et al., 2021) and SLS with acceleration (SLS-acc) (Vaswani et al., 2019); L^4Mom (Rolinek & Martius, 2018); AdSGD (Malitsky & Mishchenko, 2020); and Adam (Kingma & Ba, 2015). To eliminate the influence of randomness, we repeat the experiments 5 times with different seeds and report the averaged results. The over-parameterized deep neural networks satisfy interpolation (Zhang et al., 2021). In all Polyak-based algorithms, we use $f^*_{S_k} = 0$ throughout.

5.2.1. Results on CIFAR10 and CIFAR100

We consider the benchmark experiments for CIFAR10 and CIFAR100 with two standard image-classification architectures: 28×10 wide residual network (WRN) (Zagoruyko & Komodakis, 2016) and DenseNet121 (Huang et al., 2017), without implementation of weight-decay. The maximum epochs call is 200 and the batch size is 128. For the space concern, the details of the parameters are shown in Appendix C.3. The results on CIFAR10 and Tiny-ImageNet

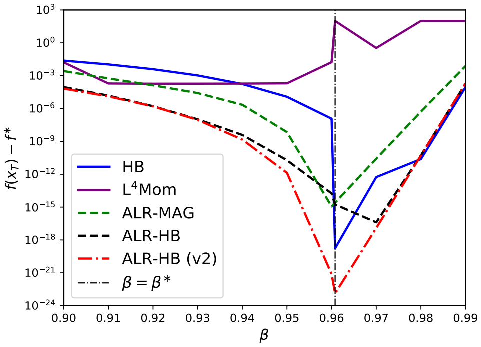


Figure 4. Least-squares under different β

Table 1. The results of CIFAR100 on WRN and DenseNet

METHOD	WRN-28-10	DENSENET121
	TEST ACCURACY (%)	
SGDM-CONST	75.36 ± 0.28	74.20 ± 1.56
ADAM	72.70 ± 0.19	72.08 ± 0.15
L4MOM	66.84 ± 0.60	67.55 ± 0.32
SPS_max	74.39 ± 0.50	73.97 ± 0.87
SLS-ACC	75.74 ± 0.19	74.81 ± 0.26
ADSGD	75.71 ± 0.29	74.72 ± 0.40
ALR-SHB	76.36 ± 0.15	74.50 ± 0.95
ALR-SMAG	$\mathbf{76.51 \pm 0.32}$	$\mathbf{75.25 \pm 0.49}$
SGDM-STEP	76.49 ± 0.37	75.12 ± 0.32

are presented in Appendix C.3 and C.4, respectively.

First, we report the results of CIFAR100 on WRN-28-10 and DenseNet121 in Figure 5 and Table 1. From Figure 5, we observe that ALR-SMAG and ALR-SHB result in the best training loss and achieves the highest accuracy. Table 1 shows that our algorithms ALR-SMAG and ALR-SHB perform better than the adaptive step size methods SPS_max, L^{4}Mom, SLS-acc and AdSGD, and are comparable to SGDM with step-decay step size (denoted by SGDM-step). Note that L^{4}Mom failed in one run of the experiment but we still report the averaged results from the 4 successful runs.

In this experiment, we borrow the idea of warmup from (Vaswani et al., 2017) to update the upper bound $\eta_{\max}$ of ALR-SHB and ALR-SMAG as $\eta_{\max} = \eta_0 \min(10^{-4}k, 1)$. The warmup heuristic has been used to mitigate the issue of converging to bad local minima for many optimization methods. The averaged result of test accuracy for each algorithm is reported in Table 2. When we incorporate the warmup (WP) technique for the maximal learning rate, our algorithms outperform SGDM with step-decay. The performance of ALR-SMAG shown in Figure 10 is insensitive to the hyper-parameter c, see Appendix C.3.

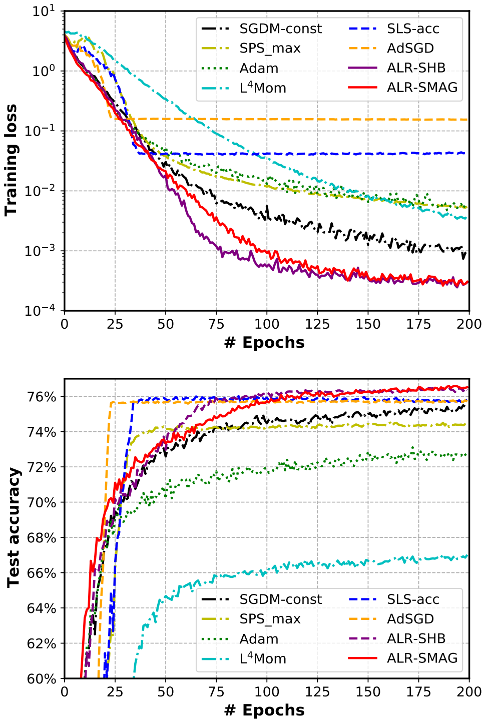


Figure 5. CIFAR100 - WRN-28-10: training loss (left) and test accuracy (right)

Table 2. Results of different step size policies under warmup

METHOD	WRN-28-10	DENSENET121
	TEST ACCURACY (%)	
SGDM-CONST + WP	75.77 ± 0.48	74.4 ± 0.47
ALR-SHB + WP	77.57 ± 0.42	77.03 ± 0.35
ALR-SMAG + WP	$\mathbf{77.63 \pm 0.21}$	$\mathbf{77.24 \pm 0.16}$
SGDM-STEP + WP	77.27 ± 0.26	76.89 ± 0.28

5.3. Enabling Weight-Decay to Improve Generalization

In neural network training, it is often desirable to incorporate weight-decay (ℓ_2-regularization) to improve generalization. It is, therefore, important to make our step sizes efficient also in this setting. However, the typical way of adding ℓ_2 regularization ($f + \frac{\lambda}{2} \|\cdot\|^2$) to the objective function is not applicable for Polyak-based algorithms because the corresponding $f^*_{S_k}$ is often inaccessible or expensive to compute. ALI-G (Berrada et al., 2020) incorporates regularization as a constraint on the feasible domain. However, promoting regularization as a constraint does not work well for our step sizes. Instead, we use a similar idea as Loshchilov & Hutter (2017) and decouple the loss and regularization terms. In ALR-SMAG, this is done by adding λx_k to the updated direction d_k and use the search direction $d_k + \lambda x_k$; see Algorithm 3 of Appendix C.5 where $\lambda > 0$ is the parameter of weight-decay. In this way, we still set $f^*_{S_k}$ to be zero because nothing changes in the networks.

We test the performance of ALR-SMAG with *weight-decay* on CIFAR100 with WRN-28-10 and compare with other state-of-the-art algorithms: AdamW under step-decay step size (denoted by AdamW-step) with $\lambda = 0.0001$; SGDM under warmup (SGDM + WP), step-decay (SGDM-step), and cosine (SGDM-cosine) step sizes with $\lambda = 0.0005$; and ALI-G (Berrada et al., 2020) with and without Nesterov momentum. For ALR-SMAG with weight-decay, we set $\lambda = 0.0005$ and $c = 0.3$. We train for 200 epochs and use batch size 128. More details are given in Appendix C.5.

The results are shown in Table 3. In addition to the best test accuracy, we also record the results at 60, 120, and 180 epochs. We observe that ALR-SMAG is able to reach a relatively high accuracy at 120 epochs. But after 120 epochs, the training process is basically saturated and the accuracy does not improve much (it even drops a little). Since c is less than 1, the step size of ALR-SMAG is still aggressive. As a result, the iteration oscillates locally, and it is difficult to converge to a certain point. In order to converge, in the final training phase (last 20% of iterations) of ALR-SMAG, we introduce a fine-tuning phase that increases c exponentially. From the last row in Table 3 we see that fine-tuning (FT) does improve the solution accuracy.

Table 3. CIFAR100 - WRN-28-10 with weight-decay (WD)

Method	Test accuracy (%)			
	#60	#120	#180	Best
AdamW-step	70.60	75.28	75.93	76.19 ± 0.13
ALI-G	**72.27**	72.83	73.12	73.40 ± 0.22
ALI-G + Mom	67.06	78.99	79.88	80.21 ± 0.14
SGDM + WP	69.88	72.02	72.32	73.47 ± 0.40
SGDM-step	60.15	75.38	80.91	81.22 ± 0.16
SGDM-cosine	64.52	71.17	81.42	81.85 ± 0.19
ALR-SMAG	63.64	**80.20**	80.69	81.18 ± 0.28
ALR-SMAG + FT	63.64	**80.20**	**81.64**	**81.89** ± 0.20

6. Conclusion

We proposed a novel approach for generalizing the popular Polyak step size to first-order methods with momentum. The resulting algorithms are significantly better than the original heavy-ball method and gradient descent with Polyak step size if the condition number is inaccessible. We demonstrated our methods are less sensitive to the choice of β than the original heavy-ball method and may avoid the instability of heavy-ball on the ill-conditioned problems. Furthermore, we extended our step sizes to the stochastic settings and demonstrated superior performance in logistic regression and deep neural network training compared to the state-of-the-art adaptive methods. In Appendix A, we extend our framework to Nesterov accelerated gradient (NAG) and provide preliminary experiments on least-squares problems. It will be interesting to study how this step size performs on a wider range of applications. Another interesting extension would be to develop techniques for adjusting the learning rate in second-order adaptive gradient methods, such as AdaGrad (Duchi et al., 2011) and Adam (Kingma & Ba, 2015) which are widely used in deep learning.

Acknowledgements

This work was supported partially by GRF 16310222, GRF 16201320, and VR 2019-05319. Xiaoyu Wang is supported by Innovation and Technology Commission of Hong Kong China under the project PRP/074/19FX.

References

Barré, M., Taylor, A., and d'Aspremont, A. Complexity guarantees for Polyak steps with momentum. In *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pp. 452–478. PMLR, 09–12 Jul 2020.

Bartlett, P., Freund, Y., Lee, W. S., and Schapire, R. E. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5): 1651–1686, 1998.

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006. doi: 10.1198/016214505000000907.

Bazaraa, M. S. and Sherali, H. D. On the choice of step size in subgradient optimization. *European Journal of Operational Research*, 7(4):380–388, 1981. doi: https://doi.org/10.1016/0377-2217(81)90096-5.

Berrada, L., Zisserman, A., and Kumar, M. P. Training neural networks for and by interpolation. In *International conference on machine learning*, pp. 799–809. PMLR, 2020.

Berrada, L., Zisserman, A., and Kumar, M. P. Comment on stochastic Polyak step-size: Performance of ALI-G. *arXiv preprint arXiv:2105.10011*, 2021.

Boyd, S., Xiao, L., and Mutapcic, A. Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2004:2004–2005, 2003.

Brännlund, U. A generalized subgradient method with relaxation step. *Mathematical Programming*, 71(2):207–219, 1995.

Camerini, P. M., Fratta, L., and Maffioli, F. On improving relaxation methods by modified gradient techniques. In *Nondifferentiable optimization*, pp. 26–34. Springer, 1975.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Freund, R. M. and Lu, H. New computational guarantees for solving convex optimization problems with first order methods, via a function growth condition measure. *Mathematical Programming*, 170(2):445–477, 2018.

Ge, R., Kakade, S. M., Kidambi, R., and Netrapalli, P. The step decay schedule: A near optimal, geometrically decaying learning rate procedure for least squares. In *Advances in Neural Information Processing Systems*, pp. 14977–14988, 2019.

Ghadimi, E., Feyzmahdavian, H. R., and Johansson, M. Global convergence of the heavy-ball method for convex optimization. In *2015 European control conference (ECC)*, pp. 310–315. IEEE, 2015.

Gower, R., Sebbouh, O., and Loizou, N. SGD for structured nonconvex functions: Learning rates, minibatching and interpolation. In *International Conference on Artificial Intelligence and Statistics*, pp. 1315–1323. PMLR, 2021.

Hardt, M., Ma, T., and Recht, B. Gradient descent learns linear dynamical systems. *Journal of Machine Learning Research*, 19:1–44, 2018.

Hazan, E. and Kakade, S. Revisiting the Polyak step size. *arXiv preprint arXiv:1905.00313*, 2019.

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.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

Karimi, H., Nutini, J., and Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 795–811. Springer, 2016.

Kingma, D. P. and Ba, J. L. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

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

Le, Y. and Yang, X. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Lenard, M. L. and Minkoff, M. Randomly generated test problems for positive definite quadratic programming. *ACM Transactions on Mathematical Software (TOMS)*, 10(1):86–96, 1984.

Liang, T. and Rakhlin, A. Just interpolate: Kernel “ridgeless” regression can generalize. *The Annals of Statistics*, 48(3):1329–1347, 2020.

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*, 2019.

Liu, Y., Gao, Y., and Yin, W. An improved analysis of stochastic gradient descent with momentum. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18261–18271, 2020.

Loizou, N., Vaswani, S., Laradji, I. H., and Lacoste-Julien, S. Stochastic Polyak step-size for SGD: An adaptive learning rate for fast convergence. In *International Conference on Artificial Intelligence and Statistics*, pp. 1306–1314. PMLR, 2021.

Loshchilov, I. and Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Ma, S., Bassily, R., and Belkin, M. The power of interpolation: Understanding the effectiveness of SGD in modern over-parameterized learning. In *International Conference on Machine Learning*, pp. 3325–3334. PMLR, 2018.

Malitsky, Y. and Mishchenko, K. Adaptive gradient descent without descent. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 6702–6712, 2020.

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

Nesterov, Y. and Polyak, B. T. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Nesterov, Y. E. A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. *Dokl. akad. nauk Sssr*, 269:543–547, 1983.

Polyak, B. *Introduction to Optimization*. Optimization Software, 1987.

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

Prazeres, M. and Oberman, A. M. Stochastic gradient descent with Polyak's learning rate. *Journal of Scientific Computing*, 89:1–16, 2021.

Robbins, H. and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

Rolinek, M. and Martius, G. L4: Practical loss-based stepsize adaptation for deep learning. In *Advances in neural information processing systems*, volume 31, 2018.

Saab, S., Phoha, S., Zhu, M., and Ray, A. An adaptive Polyak heavy-ball method. *Machine Learning*, 111(9): 3245–3277, 2022.

Sebbouh, O., Gower, R. M., and Defazio, A. Almost sure convergence rates for stochastic gradient descent and stochastic heavy ball. In *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134, pp. 3935–3971. PMLR, 2021.

Shamir, O. and Zhang, T. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International Conference on Machine Learning*, pp. 71–79, 2013.

Shor, N. Z. *Minimization methods for non-differentiable functions*, volume 3. Springer Science & Business Media, 2012.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning*, pp. 1139–1147. PMLR, 2013.

Tieleman, T. and Hinton, G. Lecture 6.5-RMSProp, coursera: Neural networks for machine learning. *Technical Report, University of Toronto*, 2012.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, volume 30, 2017.

Vaswani, S., Mishkin, A., Laradji, I., Schmidt, M., Gidel, G., and Lacoste-Julien, S. Painless stochastic gradient: Interpolation, line-search, and convergence rates. In *Advances in neural information processing systems*, volume 32, 2019.

Wang, X., Magnússon, S., and Johansson, M. On the convergence of step decay step-size for stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 34, pp. 14226–14238, 2021.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yan, Y., Yang, T., Li, Z., Lin, Q., and Yang, Y. A unified analysis of stochastic momentum methods for deep learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, pp. 2955–2961. AAAI Press, 2018.

Yang, T. and Lin, Q. RSG: Beating subgradient method without smoothness and strong convexity. *Journal of Machine Learning Research*, 19(6):1–33, 2018.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

A. Application to Nesterov Accelerated Gradient (NAG)

In Sections 2.1 and 2.2, we have shown how our framework applies to the heavy-ball and moving averaged gradient methods. However, the magic does not stop there. For instance, we further incorporate Nesterov accelerated gradient (NAG) (Nesterov, 1983)

$$\begin{aligned} v_{k+1} &= \beta v_k - \eta_k \nabla f(x_k + \beta v_k) \\ x_{k+1} &= x_k + v_{k+1} \end{aligned}$$

into the proposed framework in Section 2. In (2), let $d_k = \nabla f(x_k + \beta(x_k - x_{k-1}))$ and $\gamma = \beta$, then it reduces to the NAG algorithm. By using the convexity of f at $x_k + \beta v_k$, that is $\langle \nabla f(x_k + \beta v_k), x_k + \beta v_k - x^* \rangle \geq f(x_k + \beta v_k) - f^*$, we optimize the upper bound of $\|x_{k+1} - x^*\|^2$ with respect to the step size variable η_k, it results in the adaptive step size for NAG:

$$\eta_k = \frac{f(x_k + \beta v_k) - f^*}{\|\nabla f(x_k + \beta v_k)\|^2}. \tag{14}$$

We refer to the NAG algorithm with the step size (14) as ALR-NAG. One intuitive interpretation behind the ALR-NAG algorithm is that first, you move a momentum step $\beta(x_k - x_{k-1})$ at the current point x_k, and then you stand at this new point $\tilde{x}_k = x_k + \beta(x_k - x_{k-1})$ and perform the Polyak step size along $-\nabla f(\tilde{x}_k)$.

Next, we conduct preliminary experiments to test ALR-NAG on a least-squares problem where the condition number $\kappa = 10^4$. The first interesting result on the least-squares problem shows that ALR-NAG can obtain a more accurate solution than the original NAG (Nesterov, 1983) under optimal parameters and the accelerated gradient method (AGM) (Barré et al., 2020). Barré et al. (2020) evaluate the strong convexity constant $\hat{\mu}$ by the inverse of the Polyak step size and set the momentum parameter $\beta = (\sqrt{L} - \sqrt{\hat{\mu}})/(\sqrt{L} + \sqrt{\hat{\mu}})$ for Nesterov momentum with the knowledge of the smoothness parameter L. Besides, for each $\beta \in (0, 1)$, ALR-NAG automatically adjusts the step size which makes it less sensitive to β than NAG with $1/L$ step size where L is the largest eigenvalue of the least-squares problem. Another observation is that the optimal parameter β for ALR-NAG is not consistent with the original NAG of which the theoretical optimal momentum parameter is $\beta^* = (\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$ (Nesterov, 1983) where κ is the condition number of the problem.

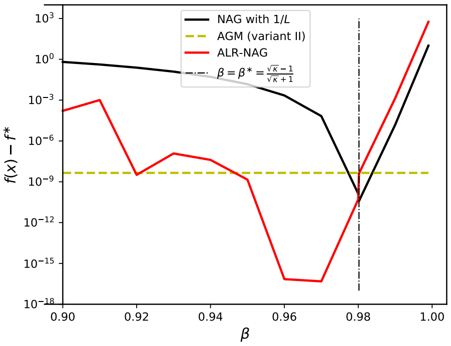


Figure 6. Results of NAG and ALR_NAG with different β on least-squares

B. Proofs of Section 4

In this section, before describing the details of the proofs, we first provide the basic definitions that omit in the main content.

Definition B.1. (Strongly convex) We say that a function f is μ-strongly convex on $\mathbb{R}^d$ if $f(y) \geq f(x) + \langle g, y - x \rangle + \frac{\mu}{2}\|y - x\|^2, \forall x, y \in \mathbb{R}^d, g \in \partial f(x)$, with $\mu > 0$.

Definition B.2. (Convex) The function f is convex on $\mathbb{R}^d$ if $f(y) \geq f(x) + \langle g, y - x \rangle$ for any $g \in \partial f(x)$ and $x, y \in \mathbb{R}^d$.

Definition B.3. (Quasar convexity (Gower et al., 2021)) Let $\zeta \in (0, 1]$ and $x^* \in \mathcal{X}^*$. A function f is ζ-quasar convex with respect to x^* if $f^* \geq f(x) + \frac{1}{\zeta}\langle g, x^* - x \rangle$ for any $x \in \mathbb{R}^d$ and $g \in \partial f(x)$.

In general, a ζ-quasar convex function f does not need to be convex. The parameter ζ controls the non-convexity of the function. If $\zeta = 1$, the quasar convex is reduced to the well-known star convexity (Nesterov & Polyak, 2006), which is a generalization of convexity. For example, $f(x) = (x^2 + \frac{1}{4})^{\frac{1}{4}}$ is quasar-convex with $\zeta = 1/2$. Learning linear dynamical systems is the practical example of a quasar-convex function and is nonconvex (Hardt et al., 2018).

In Section 4, we have provided the definition of the semi-strongly convex functions. Here we give one simple example to clarify that the semi-strongly convex function is not necessarily strongly convex. For example, $f(x) = x^2 + 2\sin(x)^2$ is semi-strongly convex with $\hat{\mu} = 2$, while the second-order derivative $\nabla^2 f(x)$ can be negative.

Definition B.4. (L-smooth) When the function f is differentiable on $\mathbb{R}^d$, we say that f is L-smooth on $\mathbb{R}^d$ if there exists a constant $L > 0$ such that $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$. This also implies that $f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|x - y\|^2$ for any $x, y \in \mathbb{R}^d$.

In this paper, we also consider a special family of non-smooth and non-strongly convex problems, whose epigraph is a polyhedron (Yang & Lin, 2018).

Definition B.5. (Polyhedral convex) For a convex minimization problem (1), suppose that the epigraph of f over $\mathcal{X}$ is a polyhedron.

The convex polyhedral problem implies the *polyhedral error condition* (Yang & Lin, 2018): there exists a constant $\kappa_1 > 0$ such that

$$\|x - x^*\| \leq \frac{1}{\kappa_1}(f(x) - f^*)$$

for all $x \in \mathcal{X}$. Some interesting applications for example ℓ_1 and ℓ_∞-constrained or regularized piece-wise linear minimization, and a submodular function minimization are polyhedral convex (Yang & Lin, 2018).

Assumption B.6. For the problem (1), we assume that (i) $\forall x_0 \in \mathbb{R}^d$, we know there exists $\delta > 0$ such that $f(x_0) - \min_{x\in\mathbb{R}^d} f(x) \leq \delta$; (ii) there exists a constant $G > 0$ such that $\max_{g\in\partial f(x)} \|g\|^2 \leq G^2$ for any $x \in \mathbb{R}^d$.

The first assumption of Assumption B.6 implies that there is a lower bound for f^*, which is also made in (Freund & Lu, 2018). This is satisfied in most machine learning applications for which we have $f^* \geq 0$. Assumption B.6(ii) is a standard assumption to be made in the non-smooth optimization (Boyd et al., 2003; Shamir & Zhang, 2013).

B.1. Proofs of Theorems and Lemmas in Section 4.1

In this part, we provide detailed proofs for the results of ALR-MAG in deterministic optimization.

Proof. **(of Lemma 4.2)**

For $k = 1$, we have $\langle d_{k-1}, x_k - x^*\rangle = \langle d_0, x_1 - x^*\rangle = 0$. For $k > 1$, suppose that $\langle d_{k-2}, x_{k-1} - x^*\rangle \geq 0$ holds, we have

$$\begin{aligned}
\langle d_{k-1}, x_k - x^*\rangle &= \langle d_{k-1}, x_k - x_{k-1} + x_{k-1} - x^*\rangle = \langle d_{k-1}, -\eta_{k-1}d_{k-1} + x_{k-1} - x^*\rangle \\
&= -\eta_{k-1}\|d_{k-1}\|^2 + \langle \nabla f(x_{k-1}) + \beta d_{k-2}, x_{k-1} - x^*\rangle \\
&= -\eta_{k-1}\|d_{k-1}\|^2 + \langle \nabla f(x_{k-1}), x_{k-1} - x^*\rangle + \beta\langle d_{k-2}, x_{k-1} - x^*\rangle \\
&\overset{(a)}{\geq} -\eta_{k-1}\|d_{k-1}\|^2 + (f(x_{k-1}) - f(x^*)) + \beta\langle d_{k-2}, x_{k-1} - x^*\rangle \geq 0.
\end{aligned}$$

where (a) follows from the convexity of f that $\langle \nabla f(x_{k-1}), x_{k-1} - x^*\rangle \geq f(x_{k-1}) - f(x^*)$ and $\eta_{k-1} \leq \frac{f(x_{k-1}) - f(x^*)}{\|d_{k-1}\|^2}$. By induction, we claim that $\langle d_{k-1}, x_k - x^*\rangle \geq 0$ for all $k \geq 1$. □

Proof. **(of Lemma 4.3)**

First, we consider the general convex functions without the smoothness assumption, then

$$\langle d_k, x_k - x^*\rangle = \langle \nabla f(x_k) + \beta d_{k-1}, x_k - x^*\rangle \geq \langle \nabla f(x_k), x_k - x^*\rangle \geq (f(x_k) - f^*) \tag{15}$$

where $\langle d_{k-1}, x_k - x^* \rangle \geq 0$ holds by Lemma 4.2. By applying Inequality (15), the distance $\|x_{k+1} - x^*\|^2$ can be estimated as:

$$\begin{aligned}
\|x_{k+1} - x^*\|^2 &= \|x_k - \eta_k d_k - x^*\|^2 = \|x_k - x^*\|^2 - 2\eta_k \langle d_k, x_k - x^* \rangle + \eta_k^2 \|d_k\|^2 \\
&\leq \|x_k - x^*\|^2 - 2\frac{(f(x_k) - f(x^*))^2}{\|d_k\|^2} + \frac{(f(x_k) - f(x^*))^2}{\|d_k\|^2} \\
&= \|x_k - x^*\|^2 - \frac{(f(x_k) - f(x^*))^2}{\|d_k\|^2} = \|x_k - x^*\|^2 - \eta_k (f(x_k) - f(x^*)) . \qquad (16)
\end{aligned}$$

As we can see, this choice of step size leads to a decrease of $\|x_{k+1} - x^*\|^2$.

Next, if the function is also L-smooth, by (Nesterov, 2003, Theorem 2.1.5), we have

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

In this case, we claim that for all $k \geq 2$

$$\langle d_{k-1}, x_k - x^* \rangle \geq \frac{1}{2L} \sum_{i=1}^{k-1} \beta^{k-1-i} \|\nabla f(x_i)\|^2 . \qquad (17)$$

When $k = 2$, we can see that

$$\begin{aligned}
\langle d_1, x_2 - x^* \rangle &= \langle \nabla f(x_1), -\eta_1 \nabla f(x_1) + x_1 - x^* \rangle = -\eta_1 \|\nabla f(x_1)\|^2 + \langle \nabla f(x_1), x_1 - x^* \rangle \\
&\geq -(f(x_1) - f^*) + \langle \nabla f(x_1), x_1 - x^* \rangle \geq \frac{1}{2L} \|\nabla f(x_1)\|^2 .
\end{aligned}$$

Then the claim (17) holds at $k = 2$. For $k > 2$, if $\langle d_{k-2}, x_{k-1} - x^* \rangle \geq \frac{1}{2L} \sum_{i=1}^{k-2} \beta^{k-2-i} \|\nabla f(x_i)\|^2$, we have

$$\begin{aligned}
\langle d_{k-1}, x_k - x^* \rangle &= \langle d_{k-1}, x_k - x_{k-1} + x_{k-1} - x^* \rangle = -\eta_{k-1} \|d_{k-1}\|^2 + \langle d_{k-1}, x_{k-1} - x^* \rangle \\
&= -\eta_{k-1} \|d_{k-1}\|^2 + \langle \nabla f(x_{k-1}), x_{k-1} - x^* \rangle + \langle \beta d_{k-2}, x_{k-1} - x^* \rangle \\
&= -\frac{f(x_{k-1}) - f^*}{\|d_{k-1}\|^2} \|d_{k-1}\|^2 + \left(f(x_{k-1}) - f^* + \frac{1}{2L} \|\nabla f(x_{k-1})\|^2 \right) + \langle \beta d_{k-2}, x_{k-1} - x^* \rangle \\
&\geq \frac{1}{2L} \|\nabla f(x_{k-1})\|^2 + \beta \frac{1}{2L} \sum_{i=1}^{k-2} \beta^{k-2-i} \|\nabla f(x_i)\|^2 = \frac{1}{2L} \sum_{i=1}^{k-1} \beta^{k-1-i} \|\nabla f(x_i)\|^2 .
\end{aligned}$$

By induction, the claim (17) is correct for all $k \geq 2$. Then applying this claim (17), we can get that

$$\begin{aligned}
\langle d_k, x_k - x^* \rangle &= \langle \nabla f(x_k) + \beta d_{k-1}, x_k - x^* \rangle \\
&\geq \left(f(x_k) - f^* + \frac{1}{2L} \|\nabla f(x_k)\|^2 \right) + \beta \frac{1}{2L} \sum_{i=1}^{k-1} \beta^{k-1-i} \|\nabla f(x_i)\|^2 \\
&= (f(x_k) - f^*) + \frac{1}{2L} \sum_{i=1}^{k} \beta^{k-i} \|\nabla f(x_i)\|^2 .
\end{aligned}$$

The distance $\|x_{k+1} - x^*\|^2$ can be evaluated as

$$\begin{aligned}
\|x_{k+1} - x^*\|^2 &= \|x_k - \eta_k d_k - x^*\|^2 = \|x_k - x^*\|^2 - 2\eta_k \langle d_k, x_k - x^* \rangle + \eta_k^2 \|d_k\|^2 \\
&\leq \|x_k - x^*\|^2 - \eta_k (f(x_k) - f^*) - 2\eta_k \frac{1}{2L} \sum_{i=1}^{k} \beta^{k-i} \|\nabla f(x_i)\|^2 \\
&\leq \|x_k - x^*\|^2 - \eta_k (f(x_k) - f^*) - \frac{1}{L} \frac{f(x_k) - f^*}{\|d_k\|^2} \sum_{i=1}^{k} \beta^{k-i} \|\nabla f(x_i)\|^2 \\
&\overset{(a)}{\leq} \|x_k - x^*\|^2 - \eta_k (f(x_k) - f^*) - \frac{(1 - \beta)}{L} (f(x_k) - f^*)
\end{aligned}$$

where (a) follows from the fact that

$$\begin{aligned}
\|d_k\|^2 &= \|\beta d_{k-1} + \nabla f(x_k)\|^2 = \beta^2 \|d_{k-1}\|^2 + \|\nabla f(x_k)\|^2 + 2\beta \langle d_{k-1}, \nabla f(x_k)\rangle \\
&\overset{(a)}{\leq} \beta^2 \|d_{k-1}\|^2 + \|\nabla f(x_k)\|^2 + \beta \left(\tau \|d_{k-1}\|^2 + \frac{1}{\tau}\|\nabla f(x_k)\|^2\right) \\
&= \beta \|d_{k-1}\|^2 + \frac{1}{1-\beta}\|\nabla f(x_k)\|^2 \overset{(b)}{\leq} \frac{1}{(1-\beta)}\sum_{i=1}^{k} \beta^{k-i}\|\nabla f(x_i)\|^2
\end{aligned} \tag{18}$$

where (a) uses the Cauchy-Schwarz inequality and we let $\tau = 1-\beta$ and (b) follows from the induction that $\|d_i\|^2 \leq \beta\|d_{i-1}\|^2 + \frac{1}{1-\beta}\|\nabla f(x_i)\|^2$ for all $i = 1, \cdots, k$ with $d_0 = 0$. Then the proof is complete. □

Theorem B.7. *(***ALR-MAG on non-smooth problems***) Consider the iterative scheme of MAG defined by (9) and the step size is selected by (10), we derive the convergence guarantees for MAG in the following cases:*

- *Suppose that the function f is convex and its gradient is bounded, then $f(x_k) - f(x^*) \to 0$ $(k \to \infty)$.*
- *If the objective function f is convex and its gradient (or subgradient) is bounded by G^2 (i.e. $\|\partial f(x)\|^2 \leq G^2$), we get that $f(\hat{x}_k) - f^* \leq \frac{G\|x_1 - x^*\|}{(1-\beta)\sqrt{k}}$ where $\hat{x}_k = \frac{1}{k}\sum_{i=1}^{k} x_i$.*
- *If the function f is μ-strongly convex and its gradient is bounded by G^2, then $\|x_k - x^*\|^2 \leq \frac{4G^2}{(1-\beta)^2\mu^2}\frac{1}{k}$.*
- *If the function is a polyhedral convex on $\mathbb{R}^d$ with $\kappa_1 > 0$ and satisfies Assumption B.6, then $\|x_k - x^*\|^2$ promotes linear convergence with a rate at least $1 - \frac{(1-\beta)^2\kappa_1^2}{G^2}$.*

Proof. **(of Theorem B.7)**

- **Convergence (suppose that convex and gradient is bounded)**: Applying the result of Lemma 4.3(i) and summing it from $k = 0, \cdots, \infty$ gives

$$\lim_{k\to\infty}\sum_{i=1}^{k}\frac{f(x_k) - f(x^*)^2}{\|d_k\|^2} \leq \|x_1 - x^*\|^2 \tag{19}$$

  Because the gradient is bounded by G^2, from (18), we have

$$\|d_k\|^2 = \|\beta d_{k-1} + \nabla f(x_k)\|^2 \leq \frac{1}{(1-\beta)}\sum_{i=1}^{k}\beta^{k-i}\|\nabla f(x_i)\|^2 \leq \frac{G^2}{(1-\beta)^2}, \tag{20}$$

  then $f(x_k) - f(x^*) \to 0$ $(k \to \infty)$.

- **If the function is only convex and gradient is bounded**, then $f(\hat{x}_k) - f(x^*) \leq \mathcal{O}(1/\sqrt{k})$. Applying the result of Lemma 4.3(i) and $\eta_k = \frac{f(x_k) - f^*}{\|d_k\|^2}$ with $\|d_k\|^2 \leq G^2/(1-\beta)^2$, we have

$$\begin{aligned}
\left(\frac{1}{k}\sum_{i=1}^{k}(f(x_i) - f(x^*))\right)^2 &\overset{(a)}{\leq} \frac{1}{k}\sum_{i=1}^{k}[f(x_i) - f(x^*)]^2 \leq \frac{G^2}{(1-\beta)^2 k}\left(\|x_1 - x^*\|^2 - \|x_{k+1} - x^*\|^2\right) \\
&\leq \frac{G^2}{(1-\beta)^2 k}\|x_1 - x^*\|^2
\end{aligned}$$

  where (a) uses the Cauchy-Schwarz inequality that $\left(\frac{1}{k}\sum_{i=1}^{k}\alpha_i\right)^2 \leq \frac{1}{k}\sum_{i=1}^{k}\alpha_i^2$ for all $\alpha_i \geq 0$. By the convexity of f, we can obtain that

$$f(\hat{x}_k) - f^* \leq \frac{1}{k}\sum_{i=1}^{k}(f(x_i) - f(x^*)) \leq \frac{G\|x_1 - x^*\|}{(1-\beta)\sqrt{k}}.$$

- **If the objective function is strongly convex and the gradient is bounded**, then $\|x_k - x^*\|^2 \leq \mathcal{O}(1/k)$.

  If the objective function is μ-strongly convex, we have $f(x_k) - f(x^*) \geq \frac{\mu}{2}\|x_k - x^*\|^2$. Due to the fact that gradient is bounded by G^2, by (20), we have $\|d_k\|^2 \leq \frac{G^2}{(1-\beta)^2}$ and

  $$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 \left(1 - \frac{(1-\beta)^2\mu^2}{4G^2}\|x_k - x^*\|^2\right)$$

  We can achieve that $\|x_k - x^*\|^2 \leq \frac{4G^2}{(1-\beta)^2\mu^2}\frac{1}{k}$ by induction.

- **If the function is polyhedral convex and Assumption B.6 holds**, in this case, we know that the polyhedral error bound condition holds: there exists a constant $\kappa_1 > 0$ such that

  $$\|x - x^*\| \leq \frac{1}{\kappa_1}(f(x) - f^*),\ \forall x \in \mathcal{X}$$

  Because the gradient (or subgradient) is bounded by G^2, that is $\max_{g\in\partial f(x_k)}\|g\|^2 \leq G^2$, from (20), we can achieve that

  $$\|d_k\|^2 \leq \frac{G^2}{(1-\beta)^2}.$$

  Applying the result of Lemma 4.3 (i) and using the definition of η_k in (10), we have

  $$\begin{aligned}\|x_{k+1} - x^*\|^2 &\leq \|x_k - x^*\|^2 - \eta_k(f(x_k) - f^*) = \|x_k - x^*\|^2 - \frac{(f(x_k) - f^*)^2}{\|d_k\|^2}\\ &\leq \|x_k - x^*\|^2 - \frac{(1-\beta)^2}{G^2}(f(x) - f^*)^2 \leq \|x_k - x^*\|^2 - \frac{(1-\beta)^2\kappa_1^2}{G^2}\|x_k - x^*\|^2\\ &\leq \left(1 - \frac{(1-\beta)^2\kappa_1^2}{G^2}\right)\|x_k - x^*\|^2.\end{aligned}$$

  In this case, $\|x_k - x^*\|^2$ promotes linear convergence with a rate at least $1 - \frac{(1-\beta)^2\kappa_1^2}{G^2}$. We must make sure that $1 - \frac{(1-\beta)^2\kappa_1^2}{G^2} > 0$. If not, we can increase G or decrease κ_1 to make the condition $1 - \frac{(1-\beta)^2\kappa_1^2}{G^2} > 0$ hold.

□

Proof. (**Proof of Theorem 4.4**)

By the convexity and smoothness, we know that Lemma 4.3(ii) holds. Then suppose that the objective function f is semi-strongly convex with $\hat{\mu}$, we can achieve that

$$\|x_{k+1} - x^*\|^2 \leq \left(1 - \frac{\hat{\mu}}{2}\left(\eta_k + \frac{(1-\beta)}{L}\right)\right)\|x_k - x^*\|^2 \leq \left(1 - \frac{\hat{\mu}(1-\beta)}{2L}\right)\|x_k - x^*\|^2$$

where $\eta_k \geq 0$. That is $\|x_k - x^*\|^2$ promotes globally linear convergence with a rate at least $(1 - (1-\beta)(2\kappa)^{-1})$ where $\kappa = L/\hat{\mu}$.

□

B.2. Proofs of Theorems in Section 4.2

We provide the essential lemmas and the proofs for the important theorems in Section 4.2. The results of ALR-SMAG for polyhedral convex and non-smooth functions and general convex functions which do not appear in the main content are presented in this part.

The first lemma follows the result of Lemma 4.2 of ALR-MAG in the deterministic case but it is more complicated.

Lemma B.8. *For convex functions, if the step size* $\eta_k \leq \frac{f_{S_k}(x_k)-f^*_{S_k}}{c\|d_k\|^2}$ *for all* $k \geq 1$*, the iterates of SMAG satisfy that* $\langle d_{k-1}, x_k - x^*\rangle \geq \left(1-\frac{1}{c}\right)\sum_{i=1}^{k-1}\beta^{k-1-i}\left(f_{S_i}(x_i)-f^*_{S_i}\right)+\sum_{i=1}^{k-1}\beta^{k-1-i}\left(f^*_{S_i}-f_{S_i}(x^*)\right)$ *for all* $k \geq 2$*.*

Proof. (**of Lemma B.8**) For $k=2$, we have

$$\begin{aligned}\langle d_1, x_2 - x^*\rangle &= \langle \nabla f_{S_1}(x_1), x_1 - \eta_1 \nabla f_{S_1}(x_1) - x^*\rangle = -\eta_1 \left\|\nabla f_{S_1}(x_1)\right\|^2 + \langle \nabla f_{S_1}(x_1), x_1 - x^*\rangle \\ &\geq -\frac{1}{c}\left(f_{S_1}(x_1)-f^*_{S_1}\right)+f_{S_1}(x_1)-f_{S_1}(x^*) = \left(1-\frac{1}{c}\right)\left(f_{S_1}(x_1)-f^*_{S_1}\right)+f^*_{S_1}-f_{S_1}(x^*).\end{aligned}$$

For $k>2$, if the claim $\langle d_{k-2}, x_{k-1}-x^*\rangle \geq \left(1-\frac{1}{c}\right)\sum_{i=1}^{k-2}\beta^{k-2-i}\left(f_{S_i}(x_i)-f^*_{S_i}\right)+\sum_{i=1}^{k-2}\beta^{k-2-i}\left(f^*_{S_i}-f_{S_i}(x^*)\right)$ holds at $k-2$, we have

$$\begin{aligned}\langle d_{k-1}, x_k - x^*\rangle &= \langle d_{k-1}, x_k - x_{k-1} + x_{k-1} - x^*\rangle = \langle d_{k-1}, -\eta_{k-1} d_{k-1} + x_{k-1} - x^*\rangle \\ &= -\eta_{k-1}\left\|d_{k-1}\right\|^2 + \left\langle \nabla f_{S_{k-1}}(x_{k-1}) + \beta d_{k-2}, x_{k-1}-x^*\right\rangle \\ &= -\eta_{k-1}\left\|d_{k-1}\right\|^2 + \left\langle \nabla f_{S_{k-1}}(x_{k-1}), x_{k-1}-x^*\right\rangle + \beta\langle d_{k-2}, x_{k-1}-x^*\rangle \\ &\overset{(a)}{\geq} -\eta_{k-1}\left\|d_{k-1}\right\|^2 + \left(f_{S_{k-1}}(x_{k-1}) - f_{S_{k-1}}(x^*)\right) + \beta\langle d_{k-2}, x_{k-1}-x^*\rangle \\ &\geq -\frac{1}{c}\left(f_{S_{k-1}}(x_{k-1})-f^*_{S_{k-1}}\right) + \left(f_{S_{k-1}}(x_{k-1})-f_{S_{k-1}}(x^*)\right) + \beta\sum_{i=1}^{k-2}\beta^{k-2-i}\left(f^*_{S_i}-f_{S_i}(x^*)\right) \\ &\quad + \beta\left(1-\frac{1}{c}\right)\sum_{i=1}^{k-2}\beta^{k-2-i}\left(f_{S_i}(x_i)-f^*_{S_i}\right) \\ &= \left(1-\frac{1}{c}\right)\sum_{i=1}^{k-1}\beta^{k-1-i}\left(f_{S_i}(x_i)-f^*_{S_i}\right)+\sum_{i=1}^{k-1}\beta^{k-1-i}\left(f^*_{S_i}-f_{S_i}(x^*)\right)\end{aligned}$$

where (a) follows from the convexity of f that $\left\langle \nabla f_{S_{k-1}}(x_k), x_k - x^*\right\rangle \geq f_{S_{k-1}}(x_k) - f_{S_{k-1}}(x^*)$ and $\eta_{k-1} \leq \frac{f_{S_{k-1}}(x_{k-1})-f^*_{S_{k-1}}}{\|d_{k-1}\|^2}$. That is to say, this claim holds at $k-1$. By induction, we can conclude that $\langle d_{k-1}, x_k - x^*\rangle \geq \left(1-\frac{1}{c}\right)\sum_{i=1}^{k-1}\beta^{k-1-i}\left(f_{S_i}(x_i)-f^*_{S_i}\right)+\sum_{i=1}^{k-1}\beta^{k-1-i}\left(f^*_{S_i}-f_{S_i}(x^*)\right)$ for all $k \geq 2$. The proof is complete. □

Proof. (**Proof of Theorem 4.6**)

In this case, the formula of step size for the stochastic version of ALR-MAG is

$$\eta_k = \min\left\{\frac{f_{S_k}(x_k)-f^*_{S_k}}{c\left\|d_k\right\|^2}, \eta_{\max}\right\}.$$

By the definition of step size, we have $\eta_k \leq \frac{f_{S_k}(x_k)-f^*_{S_k}}{c\|d_k\|^2}$. By Lemma B.8, we get that

$$\begin{aligned}\langle d_k, x_k - x^*\rangle &\geq \langle \nabla f_{S_k}(x_k), x_k - x^*\rangle + \left(1-\frac{1}{c}\right)\sum_{i=1}^{k-1}\beta^{k-i}\left(f_{S_i}(x_i)-f^*_{S_i}\right) - \sum_{i=1}^{k-1}\beta^{k-i}\left(f_{S_i}(x^*)-f^*_{S_i}\right) \\ &\geq \left(f_{S_k}(x_k)-f^*_{S_k}\right) + \left(1-\frac{1}{c}\right)\sum_{i=1}^{k-1}\beta^{k-i}\left(f_{S_i}(x_i)-f^*_{S_i}\right) - \sum_{i=1}^{k}\beta^{k-i}\left(f_{S_i}(x^*)-f^*_{S_i}\right).\end{aligned}$$

To make the analysis to be clear, we define a 0-1 event X_k. If $\eta_k = \frac{f_{S_k}(x_k)-f^*_{S_k}}{c\|d_k\|^2} \leq \eta_{\max}$, it implies that X_k happens (i.e., $X_k = 1$); otherwise, $X_k = 0$. Let $P_k = P(X_k = 1)$. First, we consider the event X_k happens, then the distance

$\|x_{k+1}-x^*\|^2$ can be estimated as

$$\begin{aligned}
\|x_{k+1}-x^*\|^2 &= \|x_k-\eta_k d_k - x^*\|^2 = \|x_k-x^*\|^2 - 2\eta_k \langle d_k, x_k - x^*\rangle + \eta_k^2\|d_k\|^2 \\
&\leq \|x_k-x^*\|^2 - 2\frac{(f_{S_k}(x_k)-f_{S_k}^*)^2}{c\|d_k\|^2} + \frac{(f_{S_k}(x_k)-f_{S_k}^*)^2}{c^2\|d_k\|^2} \\
&\quad - \frac{2}{c}\left(1-\frac{1}{c}\right)\frac{f_{S_k}(x_k)-f_{S_k}^*}{\|d_k\|^2}\sum_{i=1}^{k-1}\beta^{k-i}\left(f_{S_i}(x_i)-f_{S_i}^*\right) + 2\eta_k\sum_{i=1}^{k}\beta^{k-i}\left(f_{S_i}(x^*)-f_{S_i}^*\right) \\
&\leq \|x_k-x^*\|^2 - \frac{1}{c^2}\frac{(f_{S_k}(x_k)-f_{S_k}^*)^2}{\|d_k\|^2} - \frac{2\,(c-1)}{c^2}\frac{f_{S_k}(x_k)-f_{S_k}^*}{\|d_k\|^2}\sum_{i=1}^{k}\beta^{k-i}\left(f_{S_i}(x_i)-f_{S_i}^*\right) \\
&\quad + 2\eta_k\sum_{i=1}^{k}\beta^{k-i}\left(f_{S_i}(x^*)-f_{S_i}^*\right). \qquad (21)
\end{aligned}$$

The L-smooth property of f_{S_i} for $i=1,\cdots,k$ gives

$$\sum_{i=1}^{k}\beta^{k-i}\left(f_{S_i}(x_i)-f_{S_i}^*\right) \geq \frac{1}{2L}\sum_{i=1}^{k}\beta^{k-i}\|\nabla f_{S_i}(x_i)\|^2. \qquad (22)$$

By (18), we know that $\|d_k\|^2 \leq \frac{1}{(1-\beta)}\sum_{i=1}^{k}\beta^{k-i}\|\nabla f_{S_i}(x_i)\|^2$. Applying (22) and (18) into (21), we can achieve that

$$\begin{aligned}
\|x_{k+1}-x^*\|^2 &\leq \|x_k-x^*\|^2 - (1-\beta)\frac{(c-1)}{c^2L}\left(f_{S_k}(x_k)-f_{S_k}^*\right) + 2\eta_k\sum_{i=1}^{k}\beta^{k-i}\left(f_{S_i}(x^*)-f_{S_i}^*\right) \\
&\leq \|x_k-x^*\|^2 - (1-\beta)\frac{(c-1)}{c^2L}\left(f_{S_k}(x_k)-f_{S_k}^*\right) + 2\eta_{\max}\sum_{i=1}^{k}\beta^{k-i}\left(f_{S_i}(x^*)-f_{S_i}^*\right) \\
&\leq \|x_k-x^*\|^2 - (1-\beta)\frac{(c-1)}{c^2L}\left(f_{S_k}(x_k)-f_{S_k}(x^*)\right) + 2\eta_{\max}\sum_{i=1}^{k}\beta^{k-i}\left(f_{S_i}(x^*)-f_{S_i}^*\right) \qquad (23)
\end{aligned}$$

where $\eta_k \leq \eta_{\max}$ and $f_{S_k}^* \leq f_{S_k}(x^*)$. Taking conditional expectation w.r.t. $\mathcal{F}_k$[1] on the above inequalities gives

$$\mathbb{E}[\|x_{k+1}-x^*\|^2 \mid \mathcal{F}_k \cap \{X_k=1\}] \leq \left(1-\frac{(1-\beta)\,(c-1)\,\hat{\mu}}{2c^2L}\right)\|x_k-x^*\|^2 + \frac{2\eta_{\max}}{(1-\beta)}\sigma^2.$$

where the semi-strongly convexity of f implies that $\mathbb{E}[f_{S_k}(x_k)-f_{S_k}(x^*) \mid \mathcal{F}_k \cap \{X_k=1\}] = \mathbb{E}[f_{S_k}(x_k)-f_{S_k}(x^*) \mid \mathcal{F}_k] = f(x_k)-f(x^*) \geq \frac{\hat{\mu}}{2}\|x_k-x^*\|^2$, and we also use the Assumption 4.5 for $f_{S_i}^*$ at each step i.

If $\eta_k = \eta_{\max} < \frac{f_{S_k}(x_k)-f_{S_k}^*}{c\|d_k\|^2}$, that is $X_k=0$, we have

$$\begin{aligned}
\|x_{k+1}-x^*\|^2 &\leq \|x_k-x^*\|^2 - 2\eta_{\max}(f_{S_k}(x_k)-f_{S_k}^*) + \eta_{\max}\frac{(f_{S_k}(x_k)-f_{S_k}^*)}{c\|d_k\|^2}\|d_k\|^2 \\
&\quad + 2\eta_{\max}\sum_{i=1}^{k}\beta^{k-i}\left(f_{S_i}(x^*)-f_{S_i}^*\right) - 2\eta_{\max}\left(1-\frac{1}{c}\right)\sum_{i=1}^{k-1}\beta^{k-i}\left(f_{S_i}(x_i)-f_{S_i}^*\right) \\
&\overset{(a)}{\leq} \|x_k-x^*\|^2 - \left(2-\frac{1}{c}\right)\eta_{\max}(f_{S_k}(x_k)-f_{S_k}^*) + 2\eta_{\max}\sum_{i=1}^{k}\beta^{k-i}\left(f_{S_i}(x^*)-f_{S_i}^*\right) \\
&\overset{(b)}{\leq} \|x_k-x^*\|^2 - \left(2-\frac{1}{c}\right)\eta_{\max}(f_{S_k}(x_k)-f_{S_k}(x^*)) + 2\eta_{\max}\sum_{i=1}^{k}\beta^{k-i}\left(f_{S_i}(x^*)-f_{S_i}^*\right) \qquad (24)
\end{aligned}$$

[1] $\mathcal{F}_k$ is the σ-algebra of the set $\left\{(x_1,\nabla f_{S_1}(x_1)),\cdots,(x_{k-1},\nabla f_{S_{k-1}}(x_{k-1})),x_k\right\}$

where (a) uses the truth that $c > 1$ and $f_{S_i}(x_i) \geq f^*_{S_i}$ for each $i \geq 1$, and (b) follows from the fact that $f_{S_k}(x^*) \geq f^*_{S_k} = \min f_{S_k}(x)$. We then take the conditional expectation on the above inequality and achieve that

$$\begin{aligned}\mathbb{E}[\|x_{k+1} - x^*\|^2 \mid \mathcal{F}_k \cap \{X_k = 0\}] \leq \|x_k - x^*\|^2 - \left(2 - \frac{1}{c}\right)\eta_{\max}\mathbb{E}[(f_{S_k}(x_k) - f_{S_k}(x^*)) \mid \mathcal{F}_k \cap \{X_k = 0\}] \\ + 2\eta_{\max}\sum_{i=1}^{k}\beta^{k-i}\mathbb{E}[\left(f_{S_i}(x^*) - f^*_{S_i}\right) \mid \mathcal{F}_k \cap \{X_k = 0\}] \\ \overset{(a)}{\leq} \left(1 - \frac{(2c-1)\hat{\mu}\eta_{\max}}{2c}\right)\|x_k - x^*\|^2 + \frac{2\eta_{\max}}{(1-\beta)}\sigma^2\end{aligned}$$

where (a) uses the facts that $\mathbb{E}[f_{S_k}(x_k) - f_{S_k}(x^*) \mid \mathcal{F}_k \cap \{X_k = 0\}] = \mathbb{E}[f_{S_k}(x_k) - f_{S_k}(x^*) \mid \mathcal{F}_k] = f(x_k) - f(x^*) \geq \frac{\hat{\mu}}{2}\|x_k - x^*\|^2$ and the assumption on $f^*_{S_k}$. Overall, no matter $\frac{f_{S_k}(x_k) - f^*_{S_k}}{c\|d_k\|^2} \leq \eta_{\max}$ or not, we have

$$\begin{aligned}\mathbb{E}[\|x_{k+1} - x^*\|^2 \mid \mathcal{F}_k] &= \mathbb{E}[\|x_{k+1} - x^*\|^2 \mid \mathcal{F}_k \cap \{X_k = 1\}] + \mathbb{E}[\|x_{k+1} - x^*\|^2 \mid \mathcal{F}_k \cap \{X_k = 0\}] \\ &\leq \max\left(\mathbb{E}[\|x_{k+1} - x^*\|^2 \mid \mathcal{F}_k \cap \{X_k = 1\}], \mathbb{E}[\|x_{k+1} - x^*\|^2 \mid \mathcal{F}_k \cap \{X_k = 0\}]\right) \\ &\leq (1 - \rho_1)\|x_k - x^*\|^2 + \frac{2\eta_{\max}\sigma^2}{(1-\beta)}\end{aligned}$$

where $\rho_1 = \min\left\{\frac{(1-\beta)(c-1)\hat{\mu}}{2c^2 L}, \frac{(2c-1)\hat{\mu}\eta_{\max}}{2c}\right\}$. Telescoping the above inequality from $k = 1$ to K gives that

$$\mathbb{E}[\|x_{K+1} - x^*\|^2 \mid \mathcal{F}_K] \leq (1 - \rho_1)^K \|x_1 - x^*\|^2 + \frac{2\eta_{\max}\sigma^2}{\rho_1(1-\beta)}.$$

The proof is complete. □

Next, we consider the convergence of SMAG with (12) for the polyhedral convex functions which is a special category of nonsmooth and non-strongly convex functions.

Theorem B.9. *(**Polyhedral convex and non-smooth functions**) Under interpolation ($\sigma = 0$), we suppose that function f is polyhedral convex with $\hat{\kappa}$ and the gradient of each realization $\nabla f(x;\xi)$ is bounded by G^2. Consider the step size (12) with $c > 1/2$ and $\eta_{\max} = \infty$, we get that*

$$\mathbb{E}[\|x_{k+1} - x^*\|^2] \leq (1 - \rho_2)^k \|x_1 - x^*\|^2$$

where $\rho_2 = \frac{\hat{\kappa}^2(1-\beta)^2(2c-1)}{c^2 bG^2}$.

For the interpolated functions, Theorem B.9 generalizes the linear convergence rate beyond the smooth and semi-strongly convex functions.

Proof. (**Proof of Theorem B.9**) Under the interpolation setting, it has $\min f(x;\xi) = f^* = f(x^*)$ and all loss function f_i agrees with one common minimizer x^*. We assume that the function $f(x)$ is polyhedral convex with $\hat{\kappa} > 0$, that is $\|x - x^*\| \leq \frac{1}{\hat{\kappa}}(f(x) - f(x^*))$. Each realization function $f(x;\xi)$ is Lipschitz continuous (that is $\|\nabla f(x;\xi)\|^2 \leq G^2$ for all x). We consider the SMAG algorithm with the step size defined by (12) and $\eta_{\max} = \infty$. In this case, L-smooth property does not hold, that is to say, we can not use the Inequality (22). Due to that $\|\nabla f(x_k;\xi)\|^2 \leq G^2$ which induces that $\|\nabla f_{S_k}(x_k)\|^2 \leq G^2$. By the relationship $\|d_k\|^2 \leq \frac{1}{1-\beta}\sum_{i=1}^{k}\beta^{k-i}\|\nabla f_{S_i}(x_i)\|^2 \leq \frac{G^2}{(1-\beta)^2}$, we still can achieve that

$$\begin{aligned}\|x_{k+1} - x^*\|^2 &\leq \|x_k - x^*\|^2 - \frac{(1-\beta)^2(2c-1)}{c^2}\frac{\left(f_{S_k}(x_k) - f^*_{S_k}\right)^2}{\|d_k\|^2} + 2\eta_k\sum_{i=1}^{k}\beta^{k-i}\left(f_{S_i}(x^*) - f^*_{S_i}\right) \\ &\leq \|x_k - x^*\|^2 - (1-\beta)^2\frac{(2c-1)}{c^2}\frac{\left(f_{S_k}(x_k) - f^*_{S_k}\right)^2}{G^2} + 2\eta_k\sum_{i=1}^{k}\beta^{k-i}\left(f_{S_i}(x^*) - f^*_{S_i}\right) \\ &\overset{(a)}{=} \|x_k - x^*\|^2 - (1-\beta)^2\frac{(2c-1)}{c^2}\frac{\left(f_{S_k}(x_k) - f_{S_k}(x^*)\right)^2}{G^2} \qquad (25)\end{aligned}$$

where (a) uses the fact that $f^*_{S_k} = f_{S_k}(x^*)$ for each $k \geq 1$. Taking conditional expectation on the both side, we have

$$\begin{aligned}
\mathbb{E}[\|x_{k+1} - x^*\|^2 \mid \mathcal{F}_k] &\leq \|x_k - x^*\|^2 - \frac{(1-\beta)^2\,(2c-1)}{c^2 bG^2}\mathbb{E}[(f_{S_k}(x_k) - f_{S_k}(x^*))^2 \mid \mathcal{F}_k] \\
&\leq \|x_k - x^*\|^2 - \frac{(1-\beta)^2\,(2c-1)}{c^2 G^2}\left(\mathbb{E}[f_{S_k}(x_k) - f_{S_k}(x^*) \mid \mathcal{F}_k]\right)^2 \\
&= \|x_k - x^*\|^2 - \frac{(1-\beta)\,(2c-1)}{c^2 G^2}\left(f(x_k) - f^*]\right)^2 \\
&\leq \left(1 - \frac{\hat{\kappa}^2(1-\beta)^2\,(2c-1)}{c^2 G^2}\right)\|x_k - x^*\|^2
\end{aligned}$$

For $k = 1, \cdots, K$, we can achieve the linear convergence with a rate $\rho = 1 - \frac{\hat{\kappa}^2(1-\beta)^2(2c-1)}{c^2G^2}$. We now complete the proof. □

Theorem B.10. *(***General convex functions***) Assume that each individual function $f(x;\xi)$ is convex and L-smooth for $\xi \in \Xi$. Consider SMAG under step size (12) with $c > 1$, we can achieve that*

$$\mathbb{E}[f(\hat{x}_K) - f^*] \leq \frac{1}{Q}\frac{\|x_1 - x^*\|^2}{K} + \frac{2\eta_{\max}\sigma^2}{Q(1-\beta)}$$

where $Q = \min\left((2-1/c)\eta_{\max}, (1-\beta)(c-1)/(c^2L)\right)$ and $\hat{x} = \frac{1}{K}\sum_{k=1}^K x_k$.

The first observation is that the size of the solution's neighborhood is also proportional to $\eta_{\max}$, similar to the semi-strongly convex case of Theorem 4.6. If the interpolation condition holds, SMAG under (12) can achieve an $\mathcal{O}(1/K)$ convergence rate to reach the optimum f^*.

Proof. (**Proof of Theorem B.10**) In this case, we consider the function is convex and L-smooth.

Similar to Theorem 4.6, (23) and (24) still hold. The only difference from Theorem 4.6 is that we do not have $f(x_k) - f(x^*) \geq \frac{\mu}{2}\|x_k - x^*\|^2$. Thus

$$\begin{aligned}
\mathbb{E}[\|x_{k+1} - x^*\|^2 \mid \mathcal{F}_k] &= \mathbb{E}[\|x_{k+1} - x^*\|^2 \mid \mathcal{F}_k \cap \{X_k = 1\}] + \mathbb{E}[\|x_{k+1} - x^*\|^2 \mid \mathcal{F}_k \cap \{X_k = 0\}] \\
&\leq \max\left(\mathbb{E}[\|x_{k+1} - x^*\|^2 \mid \mathcal{F}_k \cap \{X_k = 1\}], \mathbb{E}[\|x_{k+1} - x^*\|^2 \mid \mathcal{F}_k \cap \{X_k = 0\}]\right) \\
&\leq \|x_k - x^*\|^2 - Q\left(\mathbb{E}[f_{S_k}(x_k) - f_{S_k}(x^*)\right) + 2\eta_{\max}\sum_{i=1}^k \beta^{k-i}\mathbb{E}[(f_{S_k}(x^*) - f^*_{S_k}) \mid \mathcal{F}_k] \\
&= \|x_k - x^*\|^2 - Q\left(f(x_k) - f(x^*)\right) + \frac{2\eta_{\max}\sigma^2}{1-\beta}
\end{aligned}$$

where $Q = \min\left\{\frac{(1-\beta)(c-1)}{c^2L}, \frac{(2c-1)\eta_{\max}}{c}\right\}$. Summing the above inequality from $k = 1$ to K and dividing Q to both side, we have

$$\begin{aligned}
f(\hat{x}_K) - f^* = \frac{1}{K}\sum_{k=1}^K (f(x_k) - f^*) &\leq \frac{1}{K}\sum_{k=1}^K \frac{\|x_k - x^*\|^2 - \mathbb{E}[\|x_{k+1} - x^*\|^2 \mid \mathcal{F}_k]}{Q} + \frac{2\eta_{\max}\sigma^2}{(1-\beta)Q} \\
&\leq \frac{\|x_1 - x^*\|^2}{KQ} + \frac{2\eta_{\max}\sigma^2}{(1-\beta)Q}.
\end{aligned}$$

Now, the proof is complete. □

We now investigate the convergence of ALR-SMAG for a class of nonconvex functions. The quasar convex functions with respect to $x^* \in \mathcal{X}^*$ is an extension of star convexity (Nesterov & Polyak, 2006) and convexity.

Theorem B.11. *(**Quasar convex functions**) Under interpolation ($\sigma = 0$), we assume that each individual function $f(x;\xi)$ is ζ-quasar-convex and L-smooth for $\xi \in \Xi$. Consider ALR-SMAG with $c > 1/\zeta$, we can achieve that*

$$\min_{i=1,\cdots,K} f(x_i) - f^* \leq \frac{Lc^2}{(1-\beta)(\zeta c - 1)} \frac{\|x_1 - x^*\|^2}{K}.$$

Under interpolation, Theorem B.11 provides an $\mathcal{O}(1/K)$ convergence guarantee to reach the optimum f^* for a class of nonconvex functions for ALR-SMAG.

Proof. (**Proofs of Theorem B.11**) We assume that each individual function $f(x;\xi)$ is ζ-quasar-convex and L-smooth. Under interpolation, it implies that each component function $f(x;\xi)$ agrees with a common minimizer x^*. That is to say: the mini-batch functions $f_{S_k}(x)$ is also ζ-quasar-convex and satisfies

$$\langle \nabla f_{S_k}(x_k), x_k - x^* \rangle \geq \zeta \left(f_{S_k}(x_k) - f_{S_k}^* \right)$$

where $\zeta \in (0,1]$ and $k \geq 1$. In this case, the result of Lemma B.8 is

$$\begin{aligned}
\langle d_{k-1}, x_k - x^* \rangle &\geq \left(\zeta - \frac{1}{c}\right) \sum_{i=1}^{k-1} \beta^{k-1-i} \left(f_{S_i}(x_i) - f_{S_i}^* \right) + \zeta \sum_{i=1}^{k-1} \beta^{k-1-i} \left(f_{S_i}^* - f_{S_i}(x^*) \right) \\
&= \left(\zeta - \frac{1}{c}\right) \sum_{i=1}^{k-1} \beta^{k-1-i} \left(f_{S_i}(x_i) - f_{S_i}^* \right)
\end{aligned}$$

where $\zeta > 1/c$. Then

$$\begin{aligned}
\langle d_k, x_k - x^* \rangle &\geq \langle \nabla f_{S_k}(x_k), x_k - x^* \rangle + \left(\zeta - \frac{1}{c}\right) \sum_{i=1}^{k-1} \beta^{k-i} \left(f_{S_i}(x_i) - f_{S_i}^* \right) \\
&\geq \zeta \left(f_{S_k}(x_k) - f_{S_k}^* \right) + \left(\zeta - \frac{1}{c}\right) \sum_{i=1}^{k-1} \beta^{k-i} \left(f_{S_i}(x_i) - f_{S_i}^* \right).
\end{aligned}$$

We consider the step size (12) and $\eta_{\max} = \infty$. The distance of $\|x_{k+1} - x^*\|^2$ can be evaluated as

$$\begin{aligned}
\|x_{k+1} - x^*\|^2 &= \|x_k - x^*\|^2 - 2\eta_k \langle d_k, x_k - x^* \rangle + \eta_k^2 \|d_k\|^2 \\
&\leq \|x_k - x^*\|^2 - \frac{2(f_{S_k}(x_k) - f_{S_k}^*)}{c\|d_k\|^2} \left(\zeta \left(f_{S_k}(x_k) - f_{S_k}^* \right) + \left(\zeta - \frac{1}{c}\right) \sum_{i=1}^{k-1} \beta^{k-i} \left(f_{S_i}(x_i) - f_{S_i}^* \right) \right) \\
&\quad + \eta_k \frac{(f_{S_k}(x_k) - f_{S_k}^*)}{c\|d_k\|^2} \|d_k\|^2 . \qquad (26)
\end{aligned}$$

By the smoothness property of each $f(x;\xi)$ and $\|d_k\|^2 \leq \frac{1}{1-\beta} \sum_{i=1}^{k} \beta^{k-i} \|\nabla f_{S_i}(x_i)\|^2$, we obtain that

$$\sum_{i=1}^{k-1} \beta^{k-i} \left(f_{S_i}(x_i) - f_{S_i}^* \right) \geq \frac{1}{2L} \sum_{i=1}^{k} \beta^{k-i} \|\nabla f_{S_i}(x_i)\|^2 \geq \frac{(1-\beta)}{2L} \|d_k\|^2 .$$

Incorporating the above inequality to (26) gives that

$$\begin{aligned}
\|x_{k+1} - x^*\|^2 &\leq \|x_k - x^*\|^2 - \frac{(1-\beta)(\zeta c - 1)}{Lc^2} \left(f_{S_k}(x_k) - f_{S_k}^* \right) - \left(2\zeta - \frac{1}{c}\right) \eta_k \left(f_{S_k}(x_k) - f_{S_k}^* \right) \\
&\leq \|x_k - x^*\|^2 - \frac{(1-\beta)(\zeta c - 1)}{Lc^2} \left(f_{S_k}(x_k) - f_{S_k}^* \right) \qquad (27)
\end{aligned}$$

where the last inequality holds since $\zeta > 1/c$. Taking conditional expectation w.r.t. $\mathcal{F}_k$ on the both side of (27), we achieve that

$$\mathbb{E}[\|x_{k+1} - x^*\|^2 \mid \mathcal{F}_k] \leq \|x_k - x^*\|^2 - \frac{(1-\beta)(\zeta c - 1)}{Lc^2} (f(x_k) - f^*.)$$

Diving the above inequality by a constant $Q_1 = \frac{(1-\beta)(\zeta c-1)}{Lc^2}$ and summing over $k = 1, \cdots, K$ gives that

$$\min_{i=1,\cdots,K} f(x_i) - f^* \leq \frac{1}{K}\sum_{i=1}^{K}(f(x_i) - f^*) \leq \frac{1}{Q_1}\left(\mathbb{E}[\|x_k - x^*\|^2] - \mathbb{E}[\|x_{k+1} - x^*\|^2]\right) \leq \frac{Lc^2\|x_1 - x^*\|^2}{K(1-\beta)(\zeta c - 1)}.$$

We now complete the proof. □

B.3. Theoretical Guarantees of ALR-HB on Least-Squares Problems

In this part, we consider the theoretical convergence of HB under the step size defined by (7) or (8) and get the fast linear convergence rate for ALR-HB on least-squares problems.

We recall the step size (8) that is ALR-HB(v2):

$$\eta_k = \frac{1}{2L} + \frac{f(x_k) - f^*}{\|\nabla f(x_k)\|^2} + \beta\frac{\langle \nabla f(x_k), x_k - x_{k-1}\rangle}{\|\nabla f(x_k)\|^2}.$$

In general, the step size (8) may be not positive if $\langle \nabla f(x_k), x_k - x_{k-1}\rangle \ll -(f(x_k) - f^*)$. It means that the momentum direction $x_k - x_{k-1}$ has an acute angle with $-\nabla f(x_k)$ and it also promotes the reduction on the function values, just as $-\nabla f(x_k)$. In this way, from the formula (8), the weight on $-\nabla f(x_k)$ will be reduced. However, we still choose to trust $-\nabla f(x_k)$ more which is the exact descent direction, compared to the momentum direction $x_k - x_{k-1}$. Thus, we truncate the step size to be a constant when the inner product $\langle \nabla f(x_k), x_k - x_{k-1}\rangle \leq -(f(x_k) - f^*)$. We define

$$\tilde{\eta}_k = \frac{f(x_k) - f^*}{\|\nabla f(x_k)\|^2} + \beta\frac{\langle \nabla f(x_k), x_k - x_{k-1}\rangle}{\|\nabla f(x_k)\|^2} - \frac{1-\beta}{2L}.$$

When $\langle \nabla f(x_k), x_k - x_{k-1}\rangle \geq -(f(x_k) - f^*)$, we can see that $\tilde{\eta}_k \geq 0$. Then the step size can be re-written as

$$\eta_k = \frac{1}{2L} + \frac{f(x_k) - f^*}{\|\nabla f(x_k)\|^2} + \beta\frac{\langle \nabla f(x_k), x_k - x_{k-1}\rangle}{\|\nabla f(x_k)\|^2} = \frac{2-\beta}{2L} + \tilde{\eta}_k \qquad \text{(Truncated ALR-HB(v2))}$$

If $\langle \nabla f(x_k), x_k - x_{k-1}\rangle < -(f(x_k) - f^*)$, we set $\tilde{\eta}_k = 0$ and the step size $\eta_k = \frac{2-\beta}{2L}$. In the numerical experiment on least-squares in Section 5.1, such a lower bound $(2-\beta)/(2L)$ for ALR-HB(v2) never hits. For the step size defined by (7) without L, the truncated lower bound is $(1-\beta)/(2L)$. This is a very small number for example when we set $\beta = 0.9$ which is commonly used in practice.

Theorem B.12. *(**ALR-HB(v2) for least-squares problems**) For the least-squares problem, consider the heavy-ball method defined by (6) and truncated step size by (Truncated ALR-HB(v2)), we can derive the following property:*

$$\left\|\begin{bmatrix} x_{k+1} - x^* \\ x_k - x^* \end{bmatrix}\right\|^2 = \left\|\begin{bmatrix} (1+\beta)\mathbb{I}_d - \hat{\alpha}A & -\beta\mathbb{I}_d \\ \mathbb{I}_d & \mathbf{0} \end{bmatrix}\begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \end{bmatrix}\right\|^2 - \tilde{\eta}_k^2\|\nabla f(x_k)\|^2.$$

where $\hat{\alpha} = (2-\beta)/(2L)$. Especially, if the problem is μ-strongly convex and L-smooth, we set $\beta = \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^2$ where $\kappa = L/\mu$ and $\mu = \lambda_{\min}(A), L = \lambda_{\max}(A)$, we can achieve the linear convergence rate at least

$$\left\|\begin{bmatrix} x_{k+1} - x^* \\ x_k - x^* \end{bmatrix}\right\|^2 = \rho^k\left\|\begin{bmatrix} x_2 - x^* \\ x_1 - x^* \end{bmatrix}\right\|^2.$$

where $\rho = 1 - \frac{4-\sqrt{15}}{2(\sqrt{\kappa}+1)}$.

Proof. (**of Theorem B.12**) We consider the least-squares problem,

$$f(x) = \frac{1}{2}x^T Ax + \langle x, b\rangle + c = \frac{1}{2}\|x - x^*\|_A^2 + f^*$$

where $A \in \mathbb{R}^{d\times d}$ is symmetric and positive definite, $x^* = -A^{-1}b$ and $f^* = -\frac{1}{2}b^TA^{-1}b+c$, and its gradient $\nabla f(x) = Ax+b = A(x-x^*)$. Recall the truncated step size of ALR-HB(v2), we let $\hat{\alpha} = \frac{2-\beta}{2L}$ and $\tilde{\eta}_k = \frac{f(x_k)-f^*}{\|\nabla f(x_k)\|^2} + \beta\frac{\langle\nabla f(x_k), x_k - x_{k-1}\rangle}{\|\nabla f(x_k)\|^2} - \frac{1-\beta}{2L}$. When $\langle\nabla f(x_k), x_k - x_{k-1}\rangle \geq -\left(f(x_k) - f^*\right)$, we can see that $\tilde{\eta}_k \geq 0$ and

$$\eta_k = \frac{1}{2L} + \frac{f(x_k)-f^*}{\|\nabla f(x_k)\|^2} + \beta\frac{\langle\nabla f(x_k), x_k - x_{k-1}\rangle}{\|\nabla f(x_k)\|^2} = \frac{2-\beta}{2L} + \tilde{\eta}_k$$

If $\langle\nabla f(x_k), x_k - x_{k-1}\rangle \leq -\left(f(x_k) - f^*\right)$, we have $\tilde{\eta}_k = 0$ and $\eta_k = \frac{2-\beta}{2L}$. The iterative formula of HB can be re-written as

$$\begin{aligned}
\begin{bmatrix} x_{k+1} - x^* \\ x_k - x^* \end{bmatrix} &= \begin{bmatrix} (1+\beta)\mathbb{I}_d & -\beta\mathbb{I}_d \\ \mathbb{I}_d & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \end{bmatrix} - \eta_k \begin{bmatrix} \nabla f(x_k) \\ \mathbf{0} \end{bmatrix} \\
&= \begin{bmatrix} (1+\beta)\mathbb{I}_d - \hat{\alpha}A & -\beta\mathbb{I}_d \\ \mathbb{I}_d & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \end{bmatrix} - \tilde{\eta}_k \begin{bmatrix} \nabla f(x_k) \\ \mathbf{0} \end{bmatrix}.
\end{aligned}$$

Then

$$\begin{aligned}
\left\|\begin{bmatrix} x_{k+1} - x^* \\ x_k - x^* \end{bmatrix}\right\|^2 &= \left\|\begin{bmatrix} (1+\beta)\mathbb{I}_d - \hat{\alpha}A & -\beta\mathbb{I}_d \\ \mathbb{I}_d & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \end{bmatrix} - \tilde{\eta}_k \begin{bmatrix} \nabla f(x_k) \\ \mathbf{0} \end{bmatrix}\right\|^2 \\
&= \left\|\begin{bmatrix} (1+\beta)\mathbb{I}_d - \hat{\alpha}A & -\beta\mathbb{I}_d \\ \mathbb{I}_d & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \end{bmatrix}\right\|^2 + \tilde{\eta}_k^2 \|\nabla f(x_k)\|^2 \\
&\quad - 2\tilde{\eta}_k \begin{bmatrix} \nabla f(x_k)^T & \mathbf{0^T} \end{bmatrix} \begin{bmatrix} (1+\beta)\mathbb{I}_d - \hat{\alpha}A & -\beta\mathbb{I}_d \\ \mathbb{I}_d & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \end{bmatrix} \\
&= \left\|\begin{bmatrix} (1+\beta)\mathbb{I}_d - \hat{\alpha}A & -\beta\mathbb{I}_d \\ \mathbb{I}_d & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \end{bmatrix}\right\|^2 + \tilde{\eta}_k^2 \|\nabla f(x_k)\|^2 \\
&\quad - 2\tilde{\eta}_k \left(\langle\nabla f(x_k), x_k - x^*\rangle + \beta\langle\nabla f(x_k), x_k - x_{k-1}\rangle - \hat{\alpha}\|\nabla f(x_k)\|^2\right) \\
&\overset{(a)}{\leq} \left\|\begin{bmatrix} (1+\beta)\mathbb{I}_d - \hat{\alpha}A & -\beta\mathbb{I}_d \\ \mathbb{I}_d & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \end{bmatrix}\right\|^2 + \tilde{\eta}_k^2 \|\nabla f(x_k)\|^2 \\
&\quad - 2\tilde{\eta}_k \left(f(x_k) - f^* + \frac{1}{2L}\|\nabla f(x_k)\|^2 + \beta\langle\nabla f(x_k), x_k - x_{k-1}\rangle - \frac{2-\beta}{2L}\|\nabla f(x_k)\|^2\right) \\
&\overset{(b)}{=} \left\|\begin{bmatrix} (1+\beta)\mathbb{I}_d - \hat{\alpha}A & -\beta\mathbb{I}_d \\ \mathbb{I}_d & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \end{bmatrix}\right\|^2 - \tilde{\eta}_k^2 \|\nabla f(x_k)\|^2 \qquad (28)
\end{aligned}$$

where (a) follows from $\langle\nabla f(x_k), x_k - x^*\rangle \geq f(x_k) - f^* + \frac{1}{2L}\|\nabla f(x_k)\|^2$ and $\nabla f(x_k) = A(x_k - x^*)$, and (b) uses the formula of step size $\tilde{\eta}_k = \frac{f(x_k)-f(x^*)}{\|\nabla f(x_k)\|^2} + \beta\frac{\langle\nabla f(x_k), x_k - x_{k-1}\rangle}{\|\nabla f(x_k)\|^2} - \frac{1-\beta}{2L}$. If $\langle\nabla f(x_k), x_k - x_{k-1}\rangle \leq -\left(f(x_k) - f^*\right)$, the above result is also correct due to that $\tilde{\eta}_k = 0$. Overall, we can derive that

$$\left\|\begin{bmatrix} x_{k+1} - x^* \\ x_k - x^* \end{bmatrix}\right\|^2 = \left\|\begin{bmatrix} (1+\beta)\mathbb{I}_d - \hat{\alpha}A & -\beta\mathbb{I}_d \\ \mathbb{I}_d & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \end{bmatrix}\right\|^2 - \tilde{\eta}_k^2 \|\nabla f(x_k)\|^2. \qquad (29)$$

Let

$$y_k := \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \end{bmatrix}, \qquad D := \begin{bmatrix} (1+\beta)\mathbb{I}_d - \hat{\alpha}A & -\beta\mathbb{I}_d \\ \mathbb{I}_d & \mathbf{0} \end{bmatrix}.$$

By (29), we obtain the exponential decrease in $\|y_k\|^2$:

$$\|y_{k+1}\| \leq \|Dy_k\| \leq \left\|D^k y_1\right\| \leq \left\|D^k\right\|_2 \|y_1\| \leq \left(\rho(D) + o(1)\right)^k \|y_1\|$$

where $\rho(D)$ is the spectrum of D. In order to explicitly derive the convergence rate of ALR-HB (v2), we will turn to the eigenvalues of D. Furthermore, we can see that D is permutation-similar to a block diagonal matrix with 2×2 block D_i, that is

$$D \sim \begin{bmatrix} D_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & D_2 & \cdots & \mathbf{0} \\ \vdots & & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & D_d \end{bmatrix} \text{ where } D_i = \begin{bmatrix} 1+\beta-\hat{\alpha}\lambda_i & -\beta \\ 1 & 0 \end{bmatrix} \text{ for } i = 1, 2, \cdots, d_1.$$

Therefore, to get the eigenvalues of D, it is sufficient to compute the eigenvalues for all D_i. For any $i \in [d_1]$, the eigenvalues of the 2×2 matrix are the roots of the quadratic function:

$$L(s) := s^2 - (1+\beta-\hat{\alpha}\lambda_i)s + \beta = 0, \text{ where } \Delta_i = (1+\beta-\hat{\alpha}\lambda_i)^2 - 4\beta \tag{30}$$

where $\hat{\alpha} = (2-\beta)/(2L)$ and $L = \lambda_{\max}$. Because $\lambda_i/L \le 1$, we have $1+\beta-\hat{\alpha}\lambda_i = 1+\beta-(2-\beta)\lambda_i/(2L) \ge 1+\beta-(2-\beta)/2 = \frac{3\beta}{2} > 0$. In this case, if $\Delta_i \le 0$, it is equivalent to $1+\beta-\hat{\alpha}\lambda_i \le 2\sqrt{\beta}$. If $\beta \ge \min_i \left(1 - \sqrt{\frac{\lambda_i(L+\lambda_i)}{2L^2}}\right)^2 := \left(1-\sqrt{\frac{(1+\kappa)}{2\kappa^2}}\right)^2$, then $\Delta_i \le 0$ for each i. The best convergence rate is achieved by choosing $\beta = \hat{\beta} := \left(1-\sqrt{\frac{(1+\kappa)}{2\kappa^2}}\right)^2$. The convergence rate is linear with $\rho(D) = \sqrt{\beta}$.

In the numerical experiments, we found that $\beta = \beta^*$ performs better. What is its convergence rate if we set $\beta = \beta^* < \hat{\beta}$? That is to say: there are Δ_i for $i \in [d_1]$ such that $\Delta_i > 0$. The quadratic function $L(s)$ must have two solutions denoted by $s_1 < s_2$. Because $L(0) = \beta > 0$ and $L(1) = \hat{\alpha}\lambda_i > 0$, $s_1 s_2 = \beta > 0$, and $s_1 + s_2 = 1+\beta-\hat{\alpha}\lambda_i > 0$. We can claim that $s_1 \in (0, \beta)$ and $s_2 \in (\beta, 1)$. The worst-case convergence is decided by the value of s_2. Next, we show that if $\beta = \beta^*$,

$$\begin{aligned} s_2 &= \frac{(1+\beta) - \frac{2-\beta}{2L}\lambda_i + \sqrt{\left((1+\beta) - \frac{2-\beta}{2L}\lambda_i\right)^2 - 4\beta}}{2} \\ &= \frac{(1+\beta) - \frac{2-\beta}{2L}\lambda_i + \sqrt{(1-\beta)^2 - \frac{1}{\kappa}}}{2} \le 1 - \frac{4-\sqrt{15}}{2(\sqrt{\kappa}+1)} + o\left(\frac{1}{\sqrt{\kappa}+1}\right) \end{aligned}$$

In this case, the convergence rate is at least ρ^k where $\rho \approx 1 - \frac{4-\sqrt{15}}{2(\sqrt{\kappa}+1)} < 1$. Therefore, we have proved the linear convergence for ALR-HB(v2) with the rate at least $\rho = 1 - \frac{4-\sqrt{15}}{2(\sqrt{\kappa}+1)}$.

□

C. Supplementary Numerical Results and Details

C.1. Details of Section 5.1 for Least-Squares Problems

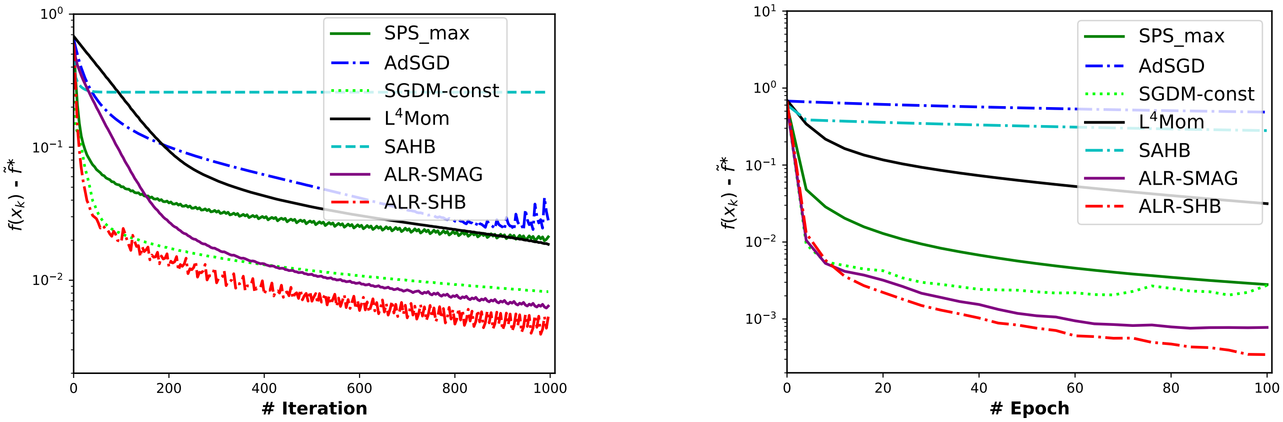
In this part, we provide the details of the experiments on the least-squares problem. The dimension $d_1 = d = 1000$ and the condition number $\kappa = 10^4$. The theoretical optimal momentum parameter $\beta^* = 0.9606$. The initial point is randomly generated and then fix it for the different test algorithms. If the step size is not specified, we select it from the set $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2\}$. For the L^{4}Mom method, we choose the momentum parameter from $\beta \in \{0.5, 0.9, 0.95, \beta^*, 0.99\}$ and the hyper-parameter $\alpha \in \{0.0001, 0.001, 0.01, 0.015, 0.1, 0.15, 1\}$ as the original paper (Rolinek & Martius, 2018).

If parameters μ and L are unknown a priori, the details of the parameters are listed below: HB with best-tuned constant step size $\eta = 0.01$ and the best-tuned momentum parameter $\beta = 0.99$ (its optimal value $\beta^* = 0.9606$); For ALR-MAG and ALR-HB, we set $\beta = 0.95$; In L^{4}Mom, we choose $\beta = 0.95$ and $\alpha = 0.01$.

C.2. Results on Logistic Regression Problems

To illustrate the practical behavior of ALR-SMAG and ALR-SHB in the convex interpolation setting, we perform experiments on logistic regression with both synthetic data and a classification dataset from LIBSVM [2]. We test our algorithms ALR-SHB and ALR-SMAG and compare with SPS_max (Loizou et al., 2021), SGDM under constant step size, AdSGD (Malitsky & Mishchenko, 2020), SAHB (Saab et al., 2022), and L^{4}Mom (Rolinek & Martius, 2018). Note that we do not estimate $f^*_{S_k}$ every iterate but set $f^*_{S_k} = 0$.

First, we follow the experiments described in section 4.1 of SPS (Loizou et al., 2021) on synthetic data for logistic regression. We do the grid search for all the parameters that are not specified and choose the best based on their practical performance. The details of the parameters in synthetic experiments are given below: (1) SPS_max (Loizou et al., 2021) with $c \in \{0.1, 0.2, 0.5, 1, 5, 10, 20\}$ and $\eta_{\max} = \{0.01, 0.1, 1, 10, 100\}$: we set $\eta_{\max} = 100$ and $c = 5$; (2) SGD with momentum (SGDM) with best tuned constant step size: $\eta \in \{0.01, 0.1, 1, 10, 100\}$ and we choose $\eta = 10$ and $\beta = 0.9$; (3) AdSGD (Malitsky & Mishchenko, 2020): $\lambda_0 = 1$ with the pair of the parameters $(\sqrt{1 + 0.01\theta}, 1/L_k)$; (4) SAHB (Saab et al., 2022): we set $\gamma_1 = 1.2, \gamma = 1, C = 10$; (5) L^{4}Mom (Rolinek & Martius, 2018), the main parameter $\alpha \in \{0.001, 0.0015, 0.01, 0.015, 0.10.15\}$ (0.15 is recommended value, but we found that $\alpha = 0.01$ works better in this case) and $\beta = 0.9$; (6) Our algorithms: $c \in \{0.1, 0.5, 1, 5, 10\}$, $\eta_{\max} = \{0.01, 0.1, 1, 10, 100\}$ and $\beta = 0.9$: for ALR-SMAG, we choose $\eta_{\max} = 100$ and $c = 5$; for ALR-SHB, we set $\eta_{\max} = 100$ and $c = 5$. The result is reported in Figure 7a. We observe that for HB and ALR-HB, the function value drops faster than other algorithms at the early stage of the training. After 400 steps, our algorithms ALR-SHB and ALR-SMAG perform better than others.



(a) Results on synthetic dataset

(b) Results on rcv1

Figure 7. Logistic regression

Similar to the synthetic dataset, we test logistic regression on a real binary classification dataset RCV1 $(n = 20242; d = 47236)$ where a 0.75 partition of the dataset is used for training, and the rest is for the test. The batch size $b = 100$ and the maximum epoch call is 100. We can see that the optimality $f(x) - \tilde{f}^*$[3] for ALR-SHB and ALR-SMAG drops faster than others. The details of the algorithms are: for SPS_max (Loizou et al., 2021), we set $\eta_{\max} = 100$ and $c = 0.5$ (recommended from their paper); We set $\eta = 10$ for SGDM and momentum parameter $\beta = 0.9$; For L^{4}Mom (Rolinek & Martius, 2018), the parameter $\alpha = 0.0015$ (0.15 is recommended value, but we found that $\alpha = 0.0015$ works better) and $\beta = 0.9$; For AdSGD (Malitsky & Mishchenko, 2020), we set $(\sqrt{(1 + 0.01\theta)}, 1/L_k)$. For SAHB (Saab et al., 2022), we set $\gamma_1 = 1, \gamma = 0.5, C = 100$. For our algorithms ALR-SHB and ALR-SMAG, we set $\eta_{\max} = 100$ and $c = 5$ for ALR-SMAG and $\eta_{\max} = 100$ and $c = 10$ for ALR-SHB.

C.3. Numerical Results on CIFAR10 and Parameters Details of Section 5.2

First, we provide the results on CIFAR10 with ResNet34 (He et al., 2016). In this experiment, we set the parameters for the tested algorithms as below: SGDM under constant step size $\eta = 0.01$; Adam with step size $\eta = 0.001$ and

[2] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

[3] $\tilde{f}^*$ is the estimation of f^* by running heavy-ball for a very long time.

$(\beta_1, \beta_2) = (0.9, 0.999)$; L^{4}Mom (Rolinek & Martius, 2018) with $\alpha = 0.01$; SPS_max (Loizou et al., 2021) with $\eta_0 = 0.1$ and $c = 0.2$ with smoothing technique to update $\eta_{\max}$; SLS-acc (Vaswani et al., 2019) with $\eta_0 = 1$ and $c = 0.1$; ALR-SHB with $c = 0.5$ and $\eta_{\max} = 0.01$ (with the warmup, under $\eta_{\max} = 0.1 \min(10^{-4}k, 1)$ and $c = 0.5$); ALR-SMAG with $c = 0.1$ and $\eta_{\max} = 0.01$ (with warmup under $\eta_{\max} = 0.1 \min(10^{-4}k, 1)$ and $c = 0.1$).

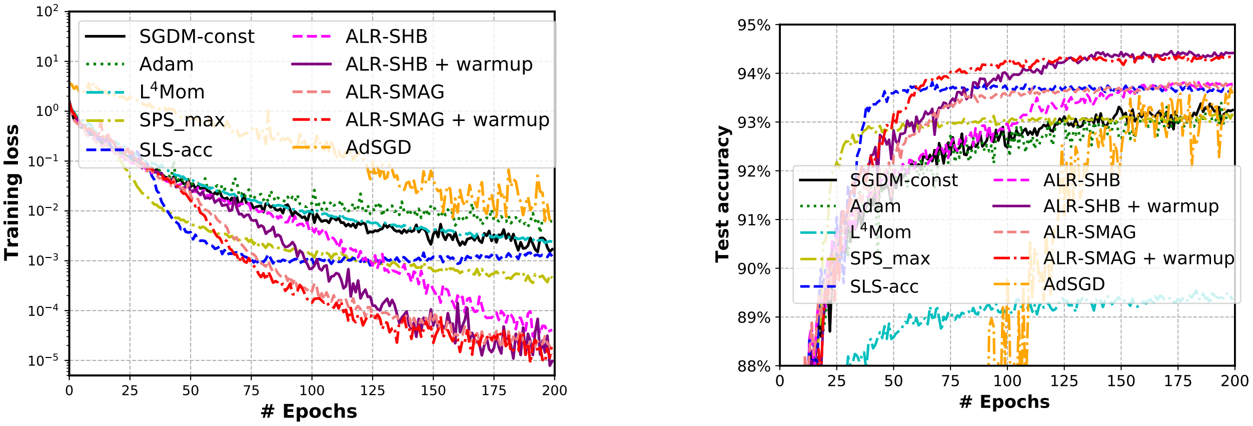


Figure 8. CIFAR10 - ResNet34: training loss (left) and test accuracy (right)

For the experiments of CIFAR100 on WRN-28-10, the details of the algorithms: SGDM under constant step size is shown below: $\eta \in \{0.001, 0.01, 0.1, 1\}$ and we set $\eta = 0.1$; SGDM with step-decay $\eta_k = \eta_0/10^{\lfloor k/K_0 \rfloor}$ where $K_0 = \lceil K/3 \rceil$ where K is the total number of iterations and we set $\eta_0 = 0.1$; Adam with $\eta = 0.001$ and $(\beta_1, \beta_2) = (0.9, 0.999)$; L^{4}Mom: we set $\alpha = 0.15$; stochastic line search with momentum (SLS-acc) (Vaswani et al., 2019) with $c = 0.1$; SPS_max (Loizou et al., 2021): we set $c = 0.2$ and $\eta_{\max} = 1$ with smoothing technique; AdSGD with parameters $(\sqrt{1 + 0.02\theta}, 1/L_k)$. For our algorithms: ALR-SHB: $\eta_{\max} = 0.1$ and $c = 0.5$, ALR-SMAG: $\eta_{\max} = 0.1$ and $c = 0.05$ (for warmup, we set $c = 0.5$ for ALR-SHB and $c = 0.05$ for ALR-SMAG, and $\eta_{\max} = \min(10^{-4}k, 1)$). In Figure 9, we present the adaptive step sizes of ALR-SMAG with and without warmup. The step size is not stably decreasing but hits the upper bound at the beginning of training, later drops for some iterations, and then hits the upper bound again in a somewhat irregular pattern.

The details of the algorithms on the experiment of CIFAR100 on DenseNet121: SGDM under constant step size $\eta = 0.01$; SGDM with step-decay $\eta_k = \eta_0/10^{\lfloor k/K_0 \rfloor}$ where $K_0 = \lceil K/3 \rceil$ and $\eta_0 = 0.01$. For our algorithms: ALR-SHB with $c = 0.5$ and $\eta_{\max} = 0.01$, ALR-SMAG with $c = 0.1$ and $\eta_{\max} = 0.01$; For warmup, we set $c = 0.5$ for ALR-SHB and $c = 0.1$ for ALR-SMAG, and $\eta_{\max} = 0.1 \min(10^{-4}k, 1)$. For the other algorithms, the parameters are the same as those on WRN-28-10.

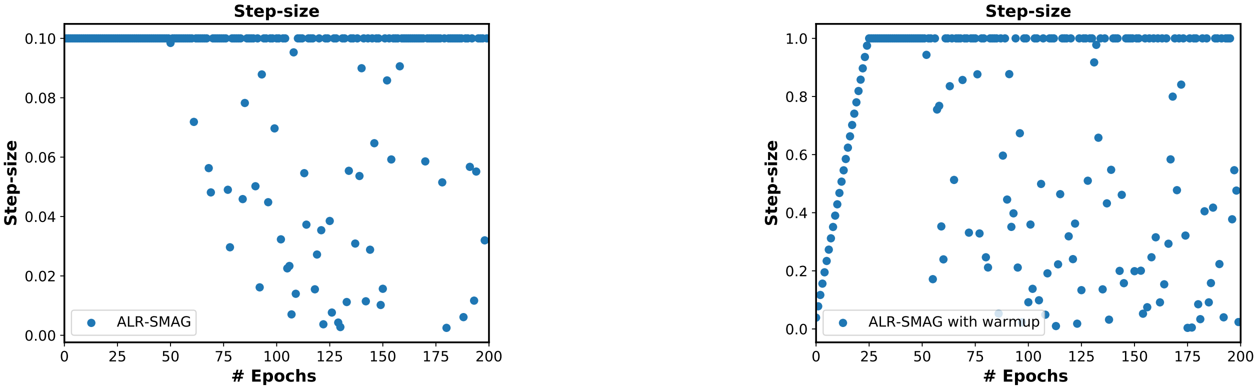


Figure 9. The plot of step sizes of ALR-SMAG and ALR-SMAG under warmup

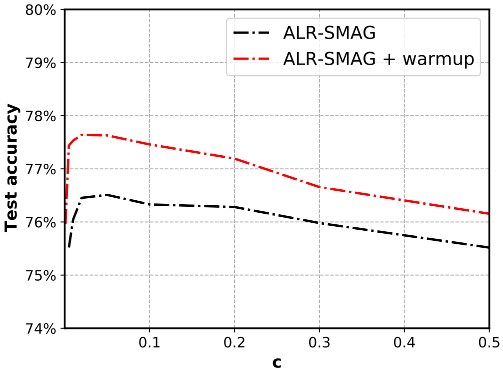


Figure 10. The behavior of the parameter c of ALR-SMAG on CIFAR100 - WRN-28-10

Finally, we show how the hyper-parameter $c > 0$ is related to the performance of ALR-SMAG. The parameter c is tested from the set $\{0.05, 0.1, 0.2, 0.3, 0.5\}$. The result is reported in Figure 10. We can see that the hyper-parameter c is insensitive to the performance of ALR-SMAG in a small range $c \in [0, 1, 0.5]$. In this case, the results for $c = 0.05, 0.1, 0.2$ are similar. We suggest that we might set the hyper-parameter c to 0.1 in the experiments on CIFARs (CIFAR10 and CIFAR100).

C.4. Results on Tiny-ImageNet200

We now turn our attention to Tiny-ImageNet200 (Le & Yang, 2015) on ResNet18 (He et al., 2016) with the pre-trained model. This dataset includes 50000 images (200 classes) for training and 10000 images for the test. In this experiment, we compare our algorithms ALR-SHB and ALR-SMAG against SGD with momentum under constant step size, the popular step-decay (Ge et al., 2019) and cosine decay (Loshchilov & Hutter, 2017) step sizes, L^{4}Mom (Rolinek & Martius, 2018) and Adam (Kingma & Ba, 2015). The results are reported in Table 4. The maximal epoch call is 200 and the batch size is 256.

The details of the algorithms are shown below: SGD with momentum (SGDM) under constant step size $\eta = 0.01$; step-decay $\eta_k = \eta_0/10^{\lfloor k/K_0 \rfloor}$ where $K_0 = \lceil K/3 \rceil$; cosine decay step size $\eta_k = 0.5\eta_0(\cos(k\pi/K)+1)$ where K is the total number of iterations and $\eta_0 = 0.01$; Adam with constant step size $\eta = 0.001$; L^{4}Mom with $\alpha = 0.15$. For our algorithm ALR-SMAG, we set $\eta_0 = 0.01$ and $c = 0.5$; with warmup, we set $\eta_{\max} = \eta_0 \min(10^{-6}k, 1)$ with $\eta_0 = 0.1$ and $c = 0.5$.

Table 4. The result of test accuracy on Tiny-ImageNet200 - ResNet18

METHOD	TEST ACCURACY (%)			
	#60	#120	#180	BEST
SGDM-CONST	64.88	65.05	65.17	65.87 ± 1.37
ADAM	58.37	58.37	58.70	59.56 ± 0.35
L^{4}MOM	65.84	65.58	65.51	66.87 ± 1.48
SGDM-STEP	65.2	**67.08**	**67.15**	67.35 ± 1.24
SGDM-COSINE	65.75	66.69	66.95	67.13 ± 1.13
ALR-SMAG	66.29	66.11	66.05	66.71 ± 1.69
ALR-SMAG + WARMUP	**67.36**	67	67.09	$\mathbf{67.66 \pm 1.05}$

For a wide range of problem classes, we can select c from a small range $c \in \{0.1, 0.5\}$. If the problem is 'difficult' to solve, i.e., necessitates a small step size, we recommend $c = 0.3$ or $c = 0.5$. For the problem at the level of training CIFARs, we can use $c = 0.1$. If a user does not have any prior information about the problems and does not want to pay any effort to tune c, we recommend using $c = 0.3$ since it works well for a wide range of problems and does not give significantly worse performance than a better-tuned value.

C.5. Details of the Experiments in Section 5.3

In the experiments for ALR-SMAG with weight-decay, the details of the algorithms are addressed as below: AdamW with step-decay step size:$\eta_0 = 0.001$ and $\eta_k = \eta_0/10^{\lfloor k/K_0 \rfloor}$ where $K_0 = \lceil K/3 \rceil$; SGDM with warmup: $\eta_k = \eta_0 \min\left(10^{-6}k, \frac{1}{\sqrt{k}}\right)$ and $\eta_0 = 0.1$; SGDM under step-decay step size $\eta_k = \eta_0/10^{\lfloor k/K_0 \rfloor}$ with $\eta_0 = 0.1$ and $K_0 = \lceil K/3 \rceil$; SGDM under cosine step size without restart $\eta_k = 0.5\eta_0(\cos(k\pi/K) + 1)$ where K is the total number of iterations and $\eta_0 = 0.1$; ALR-SMAG: $\eta_{\max} = 0.1$ and $c = 0.3$, $\lambda = 0.0005$. In the fine-tuning phase, the parameter c is exponentially increased after K_{mid} steps and $c = c_0 \exp^{\left(\frac{k-K_{mid}}{K-K_{mid}}\right)\ln(c_{\max}/c_0)}$. In this experiment, $K_{mid} = 0.8K$ and $c_{\max} = 100c_0$ where $c_0 = 0.3$.

Algorithm 3 ALR-SMAG with weight-decay

```
**Input:** $x_1, \beta \in (0,1), c > 0, \eta_{\max}, \lambda > 0, \epsilon = 10^{-5}$
**while** $x_k$ does not converge **do**
    $k \leftarrow k + 1$
    $g_k \leftarrow \frac{1}{|S_k|}\sum_{i \in S_k} \nabla f(x_k; \xi_i)$
    $f_{S_k}(x_k) \leftarrow \frac{1}{|S_k|}\sum_{i \in S_k} f(x_k; \xi_i)$
    $d_k \leftarrow \beta d_{k-1} + g_k$
    $\eta_k \leftarrow \min\left\{\eta_{\max}, \frac{f_{S_k}(x_k)}{c\|d_k\|^2 + \epsilon}\right\}$
    $x_{k+1} \leftarrow x_k - \eta_k(d_k + \boldsymbol{\lambda x_k})$
**end while**
```