Exploring the Inefficiency of Heavy Ball as Momentum Parameter Approaches 1

Xiaoge Deng , **Tao Sun*** , **Dongsheng Li*** and **Xicheng Lu**
College of Computer Science and Technology, National University of Defense Technology, China
dengxg@nudt.edu.cn, suntao.saltfish@outlook.com, {dsli, xclu}@nudt.edu.cn

Abstract

The heavy ball momentum method is a commonly used technique for accelerating training processes in the machine learning community. However, empirical evidence suggests that the convergence of stochastic gradient descent (SGD) with heavy ball may slow down when the momentum hyperparameter approaches 1. Despite this observation, there are no established theories or solutions to explain and address this issue. In this study, we provide the first theoretical result that elucidates why momentum slows down SGD as it tends to 1. To better understand this inefficiency, we focus on the quadratic convex objective in the analysis. Our findings show that momentum accelerates SGD when the scaling parameter is not very close to 1. Conversely, when the scaling parameter approaches 1, momentum impairs SGD and degrades its stability. Based on the theoretical findings, we propose a descending warmup technique for the heavy ball momentum, which exploits the advantages of the heavy ball method and overcomes the inefficiency problem when the momentum tends to 1. Numerical results demonstrate the effectiveness of the proposed SHB-DW algorithm.

1 Introduction

Training a given machine learning model $y = g(\mathbf{x}, \mathbf{w})$, parameterized by $\mathbf{w}$, is typically formulated as solving the following empirical risk minimization (ERM) problem

$$\min_{\mathbf{w}\in\mathbb{R}^d} R_S(\mathbf{w}) := \frac{1}{n}\sum_{i=1}^{n} R_i(\mathbf{w}) := \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(g(\mathbf{x}_i, \mathbf{w}); y_i),$$

where $\mathcal{L}$ denotes the loss function, $(\mathbf{x}_i, y_i)$ represents the ith data-label pair, and n is the size of the training dataset S. The main workhorse for solving the ERM problem is stochastic gradient descent (SGD) [Robbins and Monro, 1951], especially when the dimension of $\mathbf{w}$ is very high, and the number of training data n is large. Starting from an initial point $\mathbf{w}_1 \in \mathbb{R}^d$, SGD iterates as follows

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \gamma \mathbf{g}^k,$$

where $\mathbf{g}^k$ is an unbiased estimate of the gradient (i.e., $\mathbb{E}[\mathbf{g}^k] = \nabla R_S(\mathbf{w}^k)$), and $\gamma > 0$ is the learning rate.

Algorithm 1 Stochastic Heavy Ball (SHB)

```
Require: parameters $\gamma > 0, 0 \le \beta < 1$
  Initialization: $\mathbf{w}^0 = \mathbf{w}^1$
  for $k = 1, 2, \ldots$
      step 1: get $\mathbf{g}^k$ being an unbiased sample of $\nabla R_S(\mathbf{w}^k)$
      step 2: update $\mathbf{w}^k$ according to (1)
  end for
```

Accelerating SGD is crucial for machine learning, and popular acceleration techniques for SGD include adaptive learning rate [Kingma and Ba, 2015], momentum [Sutskever *et al.*, 2013], and variance reduction [Johnson and Zhang, 2013]. Among these techniques, one of the most widely used momentum-based acceleration schemes is SGD with heavy ball momentum [Polyak, 1964], also referred to as the stochastic heavy-ball method (SHB) in the context of this paper (see Algorithm 1). SHB leverages the memory along the trajectory and enjoys the following simple expression

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \gamma \mathbf{g}^k + \beta(\mathbf{w}^k - \mathbf{w}^{k-1}),\ k \ge 1, \qquad (1)$$

where $0 \le \beta < 1$ is a constant momentum parameter and $\mathbf{w}^0$ can be set the same as $\mathbf{w}^1$. When $\beta = 0$, SHB (1) reduces to plain SGD, and the choice of the momentum parameter β in SHB is crucial for accelerating SGD. It is worth noting that SHB incurs negligible computational overhead compared to SGD, making it very popular in deep learning.

Although the theoretical advantage of SHB over SGD remains unclear, there is ample empirical evidence demonstrating the superiority of SHB with an appropriate β compared to SGD. For instance, studies such as [Krizhevsky *et al.*, 2009; Sutskever *et al.*, 2013; Keskar and Socher, 2017] have shown the benefits of utilizing heavy ball momentum. However, we have found that SHB may not always outperform SGD in terms of both optimization and generalization when β is very close to 1. In the subsequent sections, we will demonstrate this phenomenon through several numerical examples.

1.1 Numerical Tests: Inefficiency of Heavy Ball

In this section, we present several numerical results to show that SHB may perform worse than the vanilla SGD when the momentum parameter β in (1) gets very close to 1.

*Corresponding authors

MNIST Classification

We conduct experiments employing several machine learning models, including the ℓ_2-regularized multi-class logistic regression and a multi-layer perceptron (MLP) for MNIST classification [LeCun and Cortes, 2010]. The dataset contains 60,000 and 10,000 grayscale images in the training and test sets. Note that the ℓ_2-regularized logistic regression model is strongly convex, while the other is nonconvex. The training batch size is set to 256 for all the following tasks.

ℓ_2-regularized multi-class logistic regression. We apply both SGD and SHB with a learning rate of 0.01 and use different momentum parameters β for SHB to train the multi-class logistic regression classifier with ℓ_2-regularization (set as 0.01) on the weights (note that this regularization guarantees the loss function to be strongly convex). We first run the full batch gradient descent for 10,000 iterations with a learning rate of 0.01 to find the global minimum, denoted as $\mathbf{w}^*$, of the model. Figure 1 illustrates the behavior of the mini-batch approximation of $\mathbb{E}[R_S(\mathbf{w}^k) - R_S(\mathbf{w}^*)]$ (the expectation is obtained via averaging over five independent runs) with $\mathbf{w}^k$ obtained from SGD or SHB with different β. It is evident that SHB converges faster than SGD at the early training stage, particularly as β increases. However, a large value of β, such as 0.9, increase oscillation and deteriorates generalization. Figure 1 also shows the test loss (b) and accuracy (c) of the model trained by different optimization algorithms over iterations, affirming that the heavy ball momentum can improve generalization.

Multi-layer perceptron (MLP). We next compare the performance of SGD and SHB with different momentum hyperparameters β in training a MLP for MNIST classification. The MLP architecture used in this study follows the same structure as described in [McMahan *et al.*, 2017], consisting of two hidden layers with 512 units, and each is activated using ReLU. We run both SGD and SHB with different β for 60 epochs and employ a learning rate of 0.01. Figure 2 shows that the heavy ball momentum can accelerate SGD and improve its generalization for solving nonconvex optimization problems, and a larger β tends to yield better performance during the early training stage. However, SHB with a large β, such as 0.99, performs significantly worse than SGD or SHB with a smaller β in both optimization and generalization. Specifically, the training loss and test accuracy of SHB with $\beta = 0.99$ plateau at much worse results than that of SGD and SHB with a smaller β.

CIFAR10 Classification

We further consider using the ResNet18 [He *et al.*, 2016a] model to classify the CIFAR10 [Krizhevsky *et al.*, 2009] dataset. This dataset consists of 50,000 and 10,000 colored images in the training and test sets, respectively. We run both SGD and SHB with different β for 90 epochs, and the batch size is set to 256. For both algorithms, we use the same learning rate of 0.01. Figure 3 plots the training loss, test loss and test accuracy versus iterations. These results confirm the previous findings. Here, we notice that when $\beta = 0.9$, SHB significantly outperforms SHB with a smaller β and SGD, which resonates with the practical choice of momentum for training ResNets [He *et al.*, 2016a]. But if we use an even bigger β, such as 0.99, SHB performs worse than SGD and much worse than SHB with a smaller β.

Numerical Observations Summary: We summarize four empirical findings from the above experiments:

- When β is very close to 1, SHB exhibits slower performance compared to SGD after several iterations, and the training curve of SHB becomes highly oscillatory.
- SHB still converges faster than SGD in the early training stage, even when β is very close to 1.
- SHB with a larger β converges faster than SHB with a smaller β, provided that β is not very close to 1.
- The test accuracy of the model trained by SHB with a β that is very close to 1 is also worse than that trained with a smaller β. That is, SHB with a β close to 1 generalizes worse than SHB with a smaller β.

Although SHB has been studied in various settings, we noticed that the theoretical interpretations for the aforementioned numerical observations have not been established yet. The focus of this paper is to provide theoretical interpretations for these empirical phenomena.

1.2 Contributions

In this paper, we aim to develop a theoretical understanding of the empirical observations above. Our contribution lies in theoretically analyzing the detriment effect of the heavy ball momentum on SGD and proposing a more efficient heavy ball momentum acceleration algorithm based on this finding. Specifically, we summarize our main contributions into the following fourfold.

- When β is close to 1, we have proven that the upper bound complexity of SHB is dominated by a smaller constant geometric factor compared to SGD in the early iterations. This result elucidates why SHB outperforms SGD in the initial training stage. For more detailed information, please refer to Section 3.1.
- We demonstrate why SHB can be slower than vanilla SGD when β is very close to 1 after a certain number of iterations. Specifically, we establish that SHB has a larger lower bound in complexity for two consecutive iterates compared to SGD. This lower bound helps explain why SHB experiences slower convergence, reduced accuracy, and decreased stability when β approaches 1. Please refer to Section 3.2 for a more detailed explanation.
- When β is not near 1, we provide a theoretical explanation for why SHB with a larger value of β converges faster than SHB with a smaller value of β. More detailed information can be found in Section 3.3.
- Building upon previous understandings and insights of momentum, we introduce a descending warmup method for SHB. This technique accelerates convergence, enhances accuracy, and improves the stability of SHB. For more detailed information, please refer to Section 4.

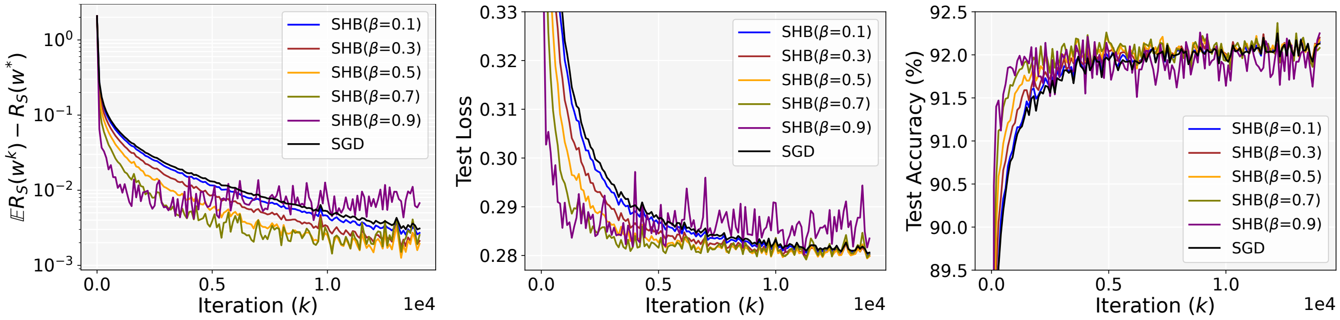

Figure 1: Comparison between the training of ℓ_2-regularized logistic regression for MNIST classification using SGD and SHB with different β. Momentum can accelerate training and improves the generalization of the regularized logistic regression model. However, as β approaches 1, SHB performs worse than SGD and SHB with a smaller β. In particular, when β is close to 1, the training curve becomes highly oscillatory, and the test accuracy also gets worse than the case when a smaller β is used.

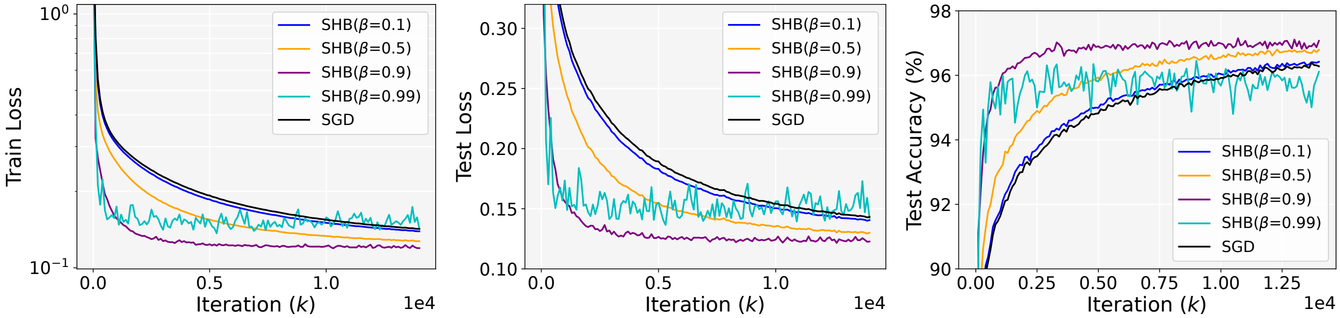

Figure 2: Comparison between the training of MLP for MNIST classification using SGD and SHB with different β. Heavy-ball momentum again can accelerate training and improve generalization, especially during the earlier training stage. However, SHB with a bigger β ($\beta = 0.99$) performs worse than SGD and SHB with smaller β values, exhibiting inferior performance in both optimization and generalization.

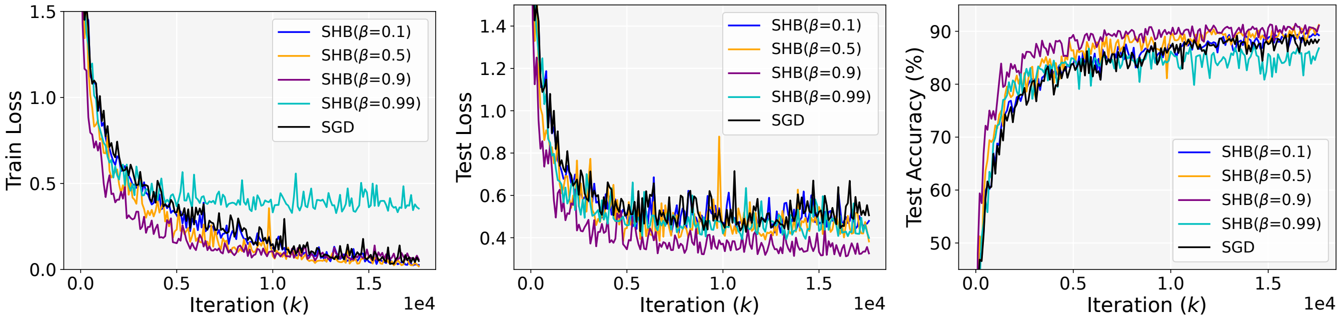

Figure 3: Comparison between training of ResNet18 for CIFAR10 classification using SGD and SHB with different β. SGD with an appropriate momentum significantly accelerates the training process and improves test accuracy. Notably, when β is set to 0.9, SHB outperforms SGD remarkably. However, if β is increased to 0.99, SHB performs worse than SGD.

1.3 Related Work

We will briefly review some of the most relevant literature regarding the development of the heavy ball momentum algorithm and its theoretical foundations. The heavy ball momentum algorithm was initially proposed by Polyak in 1964 [Polyak, 1964], and since then, its convergence properties have been extensively studied in both convex and nonconvex settings [Ochs *et al.*, 2014; Ghadimi *et al.*, 2015; Yuan *et al.*, 2016; Sun *et al.*, 2019c]. In a notable paper by Sun et al. [Sun *et al.*, 2019a], it was proven that heavy ball momentum can effectively escape saddle points with larger learning rates compared to traditional gradient descent. SGD with heavy ball momentum has become a widely used and powerful algorithm for machine learning tasks [Krizhevsky *et al.*, 2009; Sutskever *et al.*, 2013; Sun *et al.*, 2021]. Recent research efforts have focused on developing efficient heavy ball-style algorithms for training machine learning models, particularly in the field of deep learning [Yan *et al.*, 2018; Ma and Yarats, 2019; Sun *et al.*, 2019b; Gitman *et al.*, 2019; Sun *et al.*, 2020]. One notable extension of the heavy ball momentum algorithm is Nesterov's acceleration, which incorporates lookahead momentum scaled by an iteration-dependent weight [Nesterov, 1983]. This variant has proven to be efficient in accelerating gradient descent [Nesterov, 2005; Nesterov, 2013]. However, the stochastic version of Nesterov's acceleration [Wiegerinck *et al.*, 1994] faces challenges such as error accumulation and potential non-convergence [Kulunchakov and Mairal, 2019; Vaswani *et al.*, 2019; Aybat *et al.*, 2020; Wang *et al.*, 2022]. [Kidambi *et al.*, 2018; Liu and Belkin, 2020; Ganesh *et al.*, 2023] studied the inefficiency of SHB in specific cases, but failed to explain the inefficiency phenomenon when the momentum parameter approaches 1. To assess the generalization error of empirical risk minimization, [Bousquet and Elisseeff, 2002] introduced the concept of uniform stability. Building upon this notion, researchers such as [Ong, 2017; Chen *et al.*, 2018] have further investigated the bounds of generalization error for the specific cases of SHB. Additionally, [Ramezani-Kebrya *et al.*, 2024] have established a generalization bound for SHB in more general scenarios. These studies contribute to our understanding of the performance of SHB and shed light on its potential for generalization in machine learning tasks.

2 Preliminaries

2.1 Notation

We denote scalars and vectors by lowercase and bold lowercase letters, respectively, and denote matrices by uppercase boldface or uppercase curly letters. For a vector $\mathbf{x} = (x_1, \cdots, x_d) \in \mathbb{R}^d$, we denote its ℓ_2 norm by $\|\mathbf{x}\|$. The transpose of a matrix $\mathbf{A}$ is denoted as $\mathbf{A}^\top$, its spectral norm and spectral radius are denoted as $\|\mathbf{A}\|$ and $\rho(\mathbf{A})$, respectively. $\lambda_{\max}(\mathbf{A})$ and $\lambda_{\min}(\mathbf{A})$ represent the largest and smallest eigenvalues of matrix $\mathbf{A}$, respectively, and the trace of a matrix is denoted by $\mathrm{Tr}(\cdot)$.

Given two sequences $\{a_m\}$ and $\{b_m\}$, we write $a_m = \mathcal{O}(b_m)$ if there exists a positive constant $0 < C < +\infty$ such that $a_m \leq Cb_m$, and we write $a_m = \Theta(b_m)$ if there exist two positive constants C_1 and C_2 such that $a_m \leq C_1 b_m$ and $b_m \leq C_2 a_m$. We use $a_m = \tilde{\mathcal{O}}(b_m)$ and $a_m = \tilde{\Theta}(b_m)$ to hide the logarithmic factor on top of $\mathcal{O}(\cdot)$ and $\Theta(\cdot)$, respectively.

For a function $f(\mathbf{w}) : \mathbb{R}^d \to \mathbb{R}$, we denote its gradient and its Hessian as $\nabla f(\mathbf{w})$ and $\nabla^2 f(\mathbf{w})$, respectively. The minimum value of $f(\mathbf{w})$ is denoted as $\min f$, and the minimizer of $f(\mathbf{w})$ is denoted as $\mathbf{w}^*$. We use $\mathbb{E}[\cdot]$ to denote the expectation with respect to the underlying probability space.

2.2 Assumptions

Before presenting our theoretical results, we first collect several necessary assumptions for the subsequent analysis.

Assumption 1. *The stochastic gradient is an unbiased estimate of the true gradient and has bounded variance, i.e.,*

$$\mathbf{g}^k = \nabla R_S(\mathbf{w}^k) + \mathbf{e}^k \quad \textit{with} \quad \mathbb{E}[\mathbf{e}^k] = \mathbf{0},$$

and $\mathbb{E}\|\mathbf{e}^k\|^2 \leq \sigma^2$ *for some* $\sigma > 0$.

Assumption 1 is a widely accepted fundamental principle in the stochastic optimization community. It is worth noting that in the finite-sum minimization, the gradient is typically approximated using the mini-batch gradient, denoted as $\mathbf{g}^k = \left(\sum_{i=1}^m \nabla\mathcal{L}(g(\mathbf{x}_{k_i}, \mathbf{w}^k); y_{k_i})\right)/m$, where $m \ll n$ represents the batch size (commonly referred to as mini-batch SGD). This stochastic gradient is unbiased. However, relying solely on this assumption is insufficient to provide a theoretical explanation for all the experimental phenomena reported in the introduction. Therefore, to further investigate and elucidate these phenomena, additional assumptions need to be employed.

Assumption 2. *The noise* $\mathbf{e}^k$ *satisfies some distribution* $\mathcal{E}$*, whose covariance matrix*

$$\mathbb{E}[\mathbf{e}^k(\mathbf{e}^k)^\top] \equiv \Sigma \in \mathbb{R}^{d\times d}$$

is positive semi-definite and $\mathit{Tr}(\Sigma) > 0$.

In [Jastrzębski *et al.*, 2018], the authors assume that the covariance matrix of the stochastic noise is constant, and they provide an explanation for this assumption within the context of mini-batch SGD. It is important to note that Assumption 2 indicates Assumption 1, but not vice versa. A specific case of $\mathcal{E}$ in Assumption 2 is the normal Gaussian distribution, such as gradient Langevin dynamics [Welling and Teh, 2011; Stephan *et al.*, 2017; Gitman *et al.*, 2019], where Σ reduces to the identity matrix. In our paper, there is no need to assume a normal Gaussian distribution. Instead, we only require the covariance matrix of the distribution to be positive and semi-definite, which is a much weaker condition and holds for almost any distribution. The positivity of the trace of the covariance matrix ensures that $\mathbb{E}\|\mathbf{e}^k\|^2 = \mathrm{Tr}(\Sigma) > 0$.

3 Heavy Ball Slows Down the Convergence of SGD When β Is Close to 1

In this section, we analyze the convergence of SHB when R_S is quadratic and strongly convex, i.e., $\nabla^2 R_S(\mathbf{w}) \equiv \mathbf{A}$ is positive definite.

Main Techniques. First, we briefly introduce the main techniques that we used to establish the theoretical results of this study. Specifically, we reformulate SHB as follows

$$\begin{bmatrix}\mathbf{w}^{k+1} - \mathbf{w}^* \\ \mathbf{w}^k - \mathbf{w}^*\end{bmatrix} = \mathcal{T}\begin{bmatrix}\mathbf{w}^k - \mathbf{w}^* \\ \mathbf{w}^{k-1} - \mathbf{w}^*\end{bmatrix} - \gamma\begin{bmatrix}\mathbf{g}^k - \nabla R_S(\mathbf{w}^k) \\ \mathbf{0}\end{bmatrix},$$

where

$$\mathcal{T} := \begin{bmatrix}(1+\beta)\mathbf{I} - \gamma\mathbf{A} & -\beta\mathbf{I} \\ \mathbf{I} & \mathbf{0}\end{bmatrix} \in \mathbb{R}^{2d\times 2d}, \qquad (2)$$

and $\mathbf{I} \in \mathbb{R}^{d\times d}$ is the identity matrix. Let

$$\mathbf{y}^k := \begin{bmatrix}\mathbf{w}^k - \mathbf{w}^* \\ \mathbf{w}^{k-1} - \mathbf{w}^*\end{bmatrix}, \quad \text{and} \quad \mathbf{e}^k := \begin{bmatrix}\mathbf{g}^k - \nabla R_S(\mathbf{w}^k) \\ \mathbf{0}\end{bmatrix}.$$

We then have $\mathbf{y}^{k+1} = \mathcal{T}\mathbf{y}^k - \gamma\mathbf{e}^k$, and the iteration can also be formulated as the sum of a power series, given below

$$\mathbf{y}^{k+1} = \mathcal{T}^k\mathbf{y}^1 - \gamma\sum_{i=1}^k \mathcal{T}^{k-i}\mathbf{e}^i.$$

Since the noises are independent and have zero mean, taking expectations on both sides of the equation yields

$$\mathbb{E}\|\mathbf{y}^{k+1}\|^2 = \mathbb{E}\|\mathcal{T}^k\mathbf{y}^1\|^2 + \gamma^2\sum_{i=1}^k \mathbb{E}\|\mathcal{T}^{k-i}\mathbf{e}^i\|^2. \qquad (3)$$

Therefore, the problems under study turn to estimating the lower and upper bounds of $\mathbb{E}\|\mathcal{T}^{k-i}\mathbf{e}^i\|^2$. The bounded variance of the noise directly gives us

$$\mathbb{E}\|\mathcal{T}^{k-i}\mathbf{e}^i\|^2 \leq \|\mathcal{T}^{2(k-i)}\|\sigma^2.$$

The remaining problem lies in providing upper bounds of $\|\mathcal{T}^k\|$ for a fixed $k \in \mathbb{Z}^+$.

Lemma 1. *Assuming that the matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ is symmetric positive definite, and $0 < \nu \leq \lambda_{\min}(\mathbf{A})$. Let $\epsilon > 0$ be small enough and $\gamma = \Theta(\epsilon)$, and $0 \leq \beta = (1-\sqrt{\gamma\nu})^2 + \varrho < 1$ for $0 < \varrho \ll \epsilon$. We then have*

$$\|\mathcal{T}^k\| \leq \frac{C_1}{\sqrt{\gamma\nu}} \cdot (1-\sqrt{\gamma\nu})^k, \tag{4}$$

where $C_1 > 0$ is a constant independent of k and γ.

Here, the constant ϱ is introduced to ensure that $\mathcal{T}$ has distinct eigenvalues, making it similar to a diagonal matrix and simplifying the analysis. Determining the lower bound requires a non-trivial analysis, which relies on Assumption 2. Assuming that Assumption 2 holds, we can derive the following result regarding the matrix $\mathcal{T}$.

Lemma 2. *Let $\mathcal{E}$ be the stationary distribution of stochastic noise and $\xi \sim \mathcal{E} \oplus \mathbf{0}_d$. The momentum parameter satisfies $\beta = 1 - \Theta(\gamma^\tau)$ with $1 \leq \tau \leq 2$. Under Assumption 2, we can establish that*

$$\mathbb{E}\|\mathcal{T}^k\xi\|^2 \geq C_2 \cdot (1-\Theta(\gamma^\tau))^{2k}, \tag{5}$$

for some $C_2 > 0$ which is only dependent on $\mathbf{A}$.

With this in place, we can now present our main results.

3.1 Why Faster in the Early Training

Theorem 1. *Consider a quadratic function R_S with its Hessian defined as $\nabla^2 R_S(\mathbf{w}) \equiv \mathbf{A}$. The sequence $\{\mathbf{w}^k\}_{k\geq 0}$ is generated by* (1)*. Assume that $0 < \nu \leq \lambda_{\min}(\mathbf{A}) \leq \lambda_{\max}(\mathbf{A}) \leq L$ and Assumption 1 holds. For any sufficiently small $\epsilon > 0$, let $\gamma = \Theta(\epsilon)$ be chosen such that $0 \leq \beta = (1-\sqrt{\gamma\nu})^2 + \varrho < 1$, where ϱ satisfies $0 < \varrho \ll \epsilon$. Then the output of SHB satisfies*

$$\mathbb{E}\|\mathbf{w}^k - \mathbf{w}^*\|^2 \leq \frac{4C_1^2\|\mathbf{w}^1 - \mathbf{w}^*\|^2}{\gamma\nu}(1-\sqrt{\gamma\nu})^{2k} + \frac{\sqrt{\gamma}C_1^2\sigma^2}{\nu^{3/2}}.$$

Theorem 1 provides an upper bound when the momentum hyperparameter β is close to 1, and it solely relies on Assumption 1. In the early training stage, i.e., when the error $\hat{\epsilon}$ is large, we can set $\hat{\epsilon} = \sqrt{\epsilon}/\nu^{\frac{3}{2}}$, then achieving $\mathbb{E}\|\mathbf{w}^K - \mathbf{w}^*\|^2 = \mathcal{O}(\hat{\epsilon})$ for a given K only requires $(1-\sqrt{\gamma\nu})^{2K}/\gamma\nu = \mathcal{O}(\hat{\epsilon})$. Therefore, the worst-case bound for K of SHB is given by

$$K = \widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{\epsilon\nu}}\right).$$

In comparison to SGD, which requires $\widetilde{\mathcal{O}}(1/(\epsilon\nu))$ iterations to achieve the desired solution with error $\hat{\epsilon}$ in the strongly convex case [Rakhlin *et al.*, 2012], SHB significantly accelerates the convergence of SGD in the early training stage. Consequently, when the desired error is on the order of $\mathcal{O}(\hat{\epsilon})$, SHB exhibits faster convergence than SGD. Thus, even when the momentum hyperparameter β is close to 1, SHB remains faster than SGD in the early training phase for large desired errors.

Algorithm 2 Stochastic Heavy Ball with Descending Warmup (SHB-DW)

```
Require: parameters $\gamma > 0, 0 \leq \underline{\beta} < \overline{\beta} < 1, 0 < \alpha < 1$
  Initialization: $\mathbf{w}^0 = \mathbf{w}^1, \beta = \overline{\beta}$
  for $k = 1, 2, \ldots$
      step 1: update $\beta^k$ according to (7)
      step 2: get $\mathbf{g}^k$ being an unbiased sample of $\nabla R_S(\mathbf{w}^k)$
      step 3: update $\mathbf{w}^k$ as (1) with $\beta \leftarrow \beta^k$
  end for
```

3.2 Why Slower and Unstable after Enough Iterations

Theorem 2. *Let conditions of Theorem 1 and Assumption 2 hold, and $\beta = 1 - \Theta(\gamma^\tau)$ with $1 \leq \tau \leq 2$. Then the output of SHB satisfies*

$$\mathbb{E}\|\mathbf{w}^k - \mathbf{w}^*\|^2 + \mathbb{E}\|\mathbf{w}^{k-1} - \mathbf{w}^*\|^2 = \Theta(\gamma^{2-\tau}) = \Theta(\epsilon^{2-\tau}),$$

for any integer $k \geq 1$.

If $\tau > 1$, Theorem 2 indicates that no matter how many iterations are done, SHB will never achieve an error of $\mathcal{O}(\epsilon)$. On the other hand, the SGD algorithm can reach an error of $\mathcal{O}(\epsilon)$ after $\mathcal{O}(1/(\epsilon\nu))$ iterations with a suitable learning rate $\gamma = \Theta(\epsilon)$. This means that given enough iterations, SHB will be slower than SGD.

Theorem 2 also provides an explanation for the instability of the SHB curves. If $\mathbb{E}\|\mathbf{w}^K - \mathbf{w}^*\|^2 \ll \epsilon^{2-\tau}$ for some $K \in \mathbb{Z}^+$, then

$$\Theta(\epsilon^{2-\tau}) = \mathbb{E}\|\mathbf{w}^{K+1} - \mathbf{w}^*\|^2 \gg \mathbb{E}\|\mathbf{w}^K - \mathbf{w}^*\|^2.$$

3.3 Why SHB Converges Faster as β Increases Provided That β Is Not Close to 1

Given a fixed positive number β_0 such that $1 - \beta_0 \gg \epsilon$, we study the convergence of the SHB algorithm. We first present a lemma as follows.

Lemma 3. *Let $\mathbf{A} \in \mathbb{R}^{d\times d}$ be a symmetric positive definite matrix. Assume that $\epsilon > 0$ is small enough and $\gamma = \Theta(\epsilon)$. Let $0 \leq \beta \leq \beta_0 < 1$ and $1 - \beta_0 \gg \epsilon$, we can establish that*

$$\|\mathcal{T}^k\| \leq C_4 \cdot (1 - \frac{\gamma\nu}{1-\beta} + C_3\epsilon^2)^k, \tag{6}$$

where constants $C_3, C_4 > 0$ are independent of k and γ.

With Lemma 3, we can prove the convergence of SHB.

Theorem 3. *Let R_S be a quadratic function ($\nabla^2 R_S(\mathbf{w}) \equiv \mathbf{A}$), and $\{\mathbf{w}^k\}_{k\geq 0}$ be the sequence generated by* (1)*. Assuming that $0 < \nu \leq \lambda_{\min}(\mathbf{A}) \leq \lambda_{\max}(\mathbf{A}) \leq L$ and Assumption 1 holds. Given any $\epsilon > 0$ small enough, let $\gamma = \Theta(\epsilon)$ such that $0 \leq \beta \leq \beta_0 < 1$ and $1 - \beta_0 \gg \epsilon$. Then to achieve $\mathbb{E}\|\mathbf{w}^K - \mathbf{w}^*\|^2 \leq \epsilon$, the worst-case iteration complexity is*

$$K = \widetilde{\mathcal{O}}(\frac{1-\beta}{\epsilon\nu}).$$

Theorem 3 provides an explanation for why SHB can enjoy faster speed when β increases but not close to 1. As $\beta \in [0, \beta_0]$, the worst-case complexity decreases if β increases.

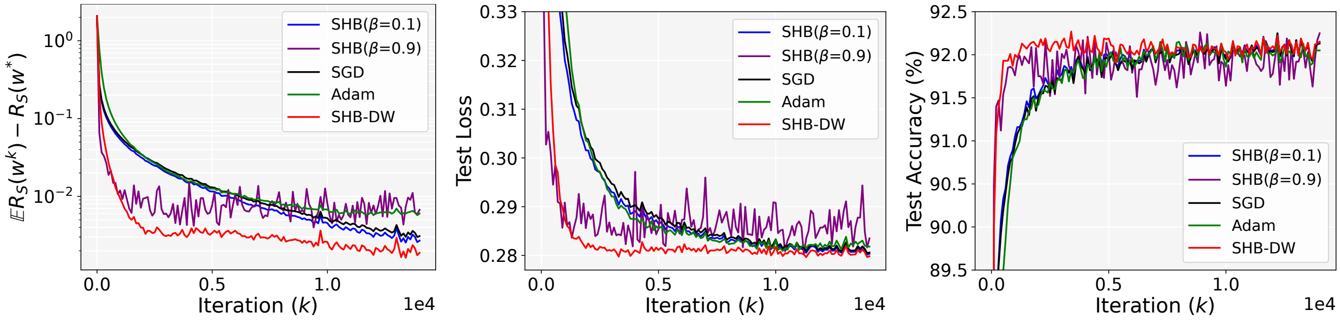

Figure 4: Comparison between training of the ℓ_2-regularized logistic regression for MNIST classification using SGD, Adam, SHB with different β, and SHB-DW. SHB-DW not only converge faster but also improves generalization.

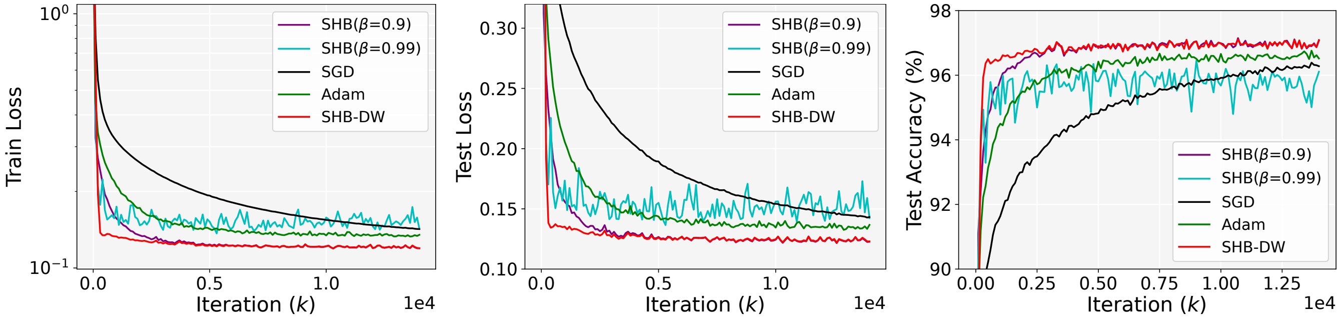

Figure 5: Comparison between training of MLP for MNIST classification using SGD, Adam, SHB with different β, and SHB-DW. SHB-DW converges faster than SHB in the earlier stage and generalizes as well as SHB with $\beta = 0.9$.

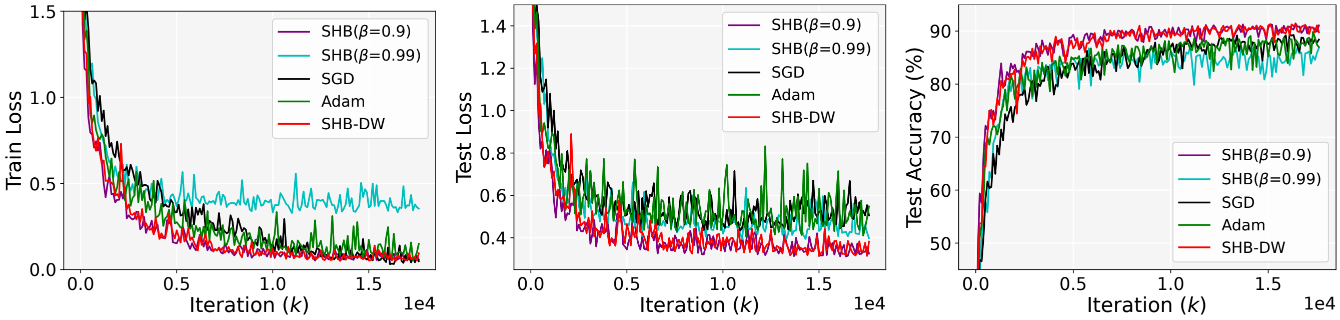

Figure 6: Comparison between training of ResNet18 for CIFAR10 classification using SGD, Adam, SHB with different β, and SHB-DW. Again, SHB-DW converges faster in the early stages compared to Adam and SGD, while also demonstrating similar generalization capabilities to SHB with $\beta = 0.9$.

4 Descending Warmup

Our results above demonstrate that SHB with a large β accelerates convergence during the early training stage but causes instability and reduces accuracy after sufficient iterations. Based on this insight, we propose an initialization method that employs a large momentum hyperparameter $\beta^0 := \overline{\beta} \in (0, 1)$, which is then reduced to a fixed number $\underline{\beta} \in [0, 1)$ according to the following rule

$$\beta^{k+1} = \begin{cases} \max\{\alpha * \beta^k, \underline{\beta}\}, & \text{if } 0 < \underline{\beta} < 1, \\ \max\{\alpha * \beta^k - 10^{-6}, \underline{\beta}\}, & \text{if } \underline{\beta} = 0, \end{cases} \tag{7}$$

where $0 < \alpha < 1$ is the descending factor. It is straightforward to see that $\{\beta^k\}_{k \geq 0}$ will be unchanged and remains as $\underline{\beta}$ after the following number of iterations

$$K(\overline{\beta}, \underline{\beta}, \alpha) := \begin{cases} \ln(\overline{\beta}/\underline{\beta}) / \ln \frac{1}{\alpha}, & \text{if } \underline{\beta} \neq 0, \\ \ln(10^6 \overline{\beta}) / \ln \frac{1}{\alpha}, & \text{if } \underline{\beta} = 0. \end{cases}$$

Therefore, (7) is actually a warmup process, and called *Descending Warmup (DW)* in this study. We formulate SHB with DW (SHB-DW) in Algorithm 2. In the early iterations of SHB-DW, relatively large momentum hyperparameters are utilized to achieve fast convergence, while smaller momentum hyperparameters are employed in the subsequent iterations to produce stable and accurate solutions.

Algorithm Model	SGD	Adam	SHB($\beta = 0.9$)	SHB-DW
ResNet18	92.66 ± 0.41	91.08 ± 0.17	93.26 ± 0.41	$\mathbf{93.38 \pm 0.52}$
ResNet34	92.18 ± 0.61	91.49 ± 0.17	92.95 ± 0.52	$\mathbf{93.22 \pm 0.41}$

Table 1: Test accuracy (%) of ResNet18 and ResNet34 for CIFAR10 classification, where the models are trained by SGD, Adam, SHB and SHB-DW algorithms. Each experiment was repeated five times for different seeds.

Let the sequence $\{\mathbf{w}^k\}_{k\geq 0}$ be generated by the SHB-DW algorithm. Since the iterates of SHB-DW after $K(\overline{\beta}, \underline{\beta}, \alpha)$ iterations can be regarded as SHB with an initialization set as $\mathbf{w}^{K(\overline{\beta},\underline{\beta},\alpha)}$, the stationary analysis of SHB-DW is then trivial and will not be repeated. To get the fine-grained convergence result for SHB-DW, we just need to bound $\|\mathbf{w}^* - \mathbf{w}^{K(\overline{\beta},\underline{\beta},\alpha)+1}\|$ under the iteration-dependent momentum hyperparameters.

5 Numerical Experiments

5.1 Descending Warmup Improves SHB in Optimization and Generalization

In this section, we replicate the experiments conducted in Section 1.1 using the proposed SHB-DW algorithm. Our theoretical findings indicate that a larger momentum parameter can be employed to speed up the training process in the early stages, and subsequently, a β value not close to 1 is required to ensure optimal performance. By default, we set the initial momentum hyperparameter β^0 to 0.999 for warmup and decrease it by a factor of $\alpha = 0.999$ after each iteration. For the ℓ_2-regularized multi-class logistic regression task, the lower bound of the momentum hyperparameter $\underline{\beta}$ for SHB-DW is set to 0.1. While for the other two tasks, Figures 2 and 3 show that SHB achieves favorable results with $\beta = 0.9$. Therefore, we set $\underline{\beta}$ of SHB-DW to 0.9 for the two tasks. Additionally, we compare the widely used Adam algorithm [Kingma and Ba, 2015]. Since Adam requires a relatively small learning rate to fully utilize its performance [Kingma and Ba, 2015], we set the learning rate of this algorithm to 0.0001, while keeping all other parameters unchanged. In Figures 4, 5, and 6, the performance of SGD, Adam, SHB ($\beta = 0.9, 0.99$), and SHB-DW is compared in the three benchmark experiments described in Section 1.1. These results confirm the superiority of SHB-DW: the decreasing warmup approach accelerates training in the early stages, and compensates for the performance degradation caused by the use of a large β.

5.2 SHB-DW for Deep Learning

To further validate the effectiveness of the descending warmup technique in training deep neural networks, we conduct more detailed experiments on CIFAR10 using ResNet18 and ResNet34 [He *et al.*, 2016b]. We compare four optimization algorithms, SGD, Adam, SHB, and SHB-DW. The momentum parameter β for SHB is set to 0.9, a commonly utilized value in practice [He *et al.*, 2016a], and the settings for SHB-DW remained the same as in Section 5.1. The algorithms are run for 200 epochs with a batch size of 256. The initial learning rate for Adam is set to 0.001, while the others are set to 0.1. All algorithms use a decreasing learning rate strategy, i.e., decreasing by a factor of 10 at the 60th, 120th and 180th epochs, respectively. Table 1 presents the test accuracy of the two models trained by the four different optimizers, and the training process is illustrated in Figure 7 (ResNet34). Compared with other optimization algorithms, SHB-DW exhibits greater stability and efficiency, outperforming SGD, Adam, and SHB in test accuracy.

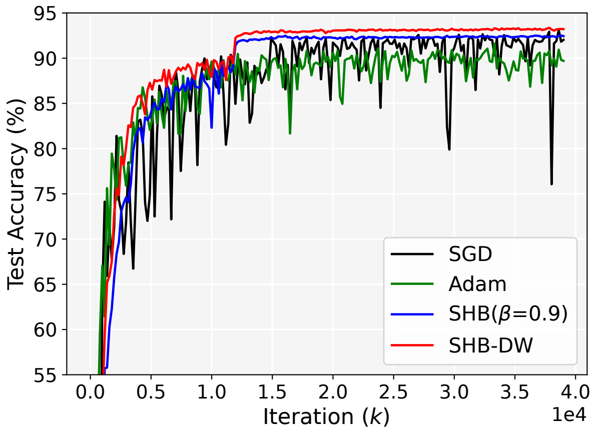

Figure 7: Test accuracy of ResNet34 for CIFAR10 classification using SGD, Adam, SHB with $\beta = 0.9$, and SHB-DW.

6 Concluding Remarks

In this paper, we have provided an explanation for the deterioration of convergence speed in the heavy ball momentum when the hyperparameter β approaches 1. Our analysis is based on novel reformulations of the heavy ball algorithm, allowing us to utilize non-Lyapunov analysis to establish the lower bound of SHB. To the best of our knowledge, our results are the first to explain how a hyperparameter close to 1 in heavy ball momentum can negatively impact convergence and lead to instability in SHB.

There are several potential directions for future research: 1) Can we extend our analysis to Nesterov's momentum, another widely utilized technique in deep neural network training? 2) Can we generalize our analysis to Nesterov's acceleration with restart, which has been empirically observed to significantly enhance the performance of SGD [Wang *et al.*, 2022]? 3) Can we establish similar theoretical results for heavy ball momentum when combined with adaptive learning rate algorithms, such as Adam?

Ethical Statement

This study focuses on the fundamental optimization algorithm and its analysis. No ethical issues are discussed or related to the research.

Acknowledgments

This work was supported in part by the National Science Foundation of China (No. 62025208, No. 62376278), Hunan Provincial Natural Science Foundation of China (No. 2022JJ10065), Young Elite Scientists Sponsorship Program by CAST (No. 2022QNRC001), Continuous Support of PDL (No. WDZC20235250101), and Xiangjiang Laboratory Fund (No. 22XJ01012).

References

[Aybat *et al.*, 2020] Necdet Serhat Aybat, Alireza Fallah, Mert Gurbuzbalaban, and Asuman Ozdaglar. Robust accelerated gradient methods for smooth strongly convex functions. *SIAM Journal on Optimization*, 30(1):717–751, 2020.

[Bousquet and Elisseeff, 2002] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.

[Chen *et al.*, 2018] Yuansi Chen, Chi Jin, and Bin Yu. Stability and convergence trade-off of iterative optimization algorithms. *arXiv preprint arXiv:1804.01619*, 2018.

[Ganesh *et al.*, 2023] Swetha Ganesh, Rohan Deb, Gugan Thoppe, and Amarjit Budhiraja. Does momentum help in stochastic optimization? A sample complexity analysis. In Robin J. Evans and Ilya Shpitser, editors, *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 602–612. PMLR, 31 Jul–04 Aug 2023.

[Ghadimi *et al.*, 2015] Euhanna Ghadimi, Hamid Reza Feyzmahdavian, and Mikael Johansson. Global convergence of the heavy-ball method for convex optimization. In *European control conference*, pages 310–315, 2015.

[Gitman *et al.*, 2019] Igor Gitman, Hunter Lang, Pengchuan Zhang, and Lin Xiao. Understanding the role of momentum in stochastic gradient methods. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[He *et al.*, 2016a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[He *et al.*, 2016b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[Jastrzębski *et al.*, 2018] Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in SGD. *Workshop of ICLR*, 2018.

[Johnson and Zhang, 2013] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[Keskar and Socher, 2017] Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from Adam to SGD. *arXiv preprint arXiv:1712.07628*, 2017.

[Kidambi *et al.*, 2018] Rahul Kidambi, Praneeth Netrapalli, Prateek Jain, and Sham Kakade. On the insufficiency of existing momentum schemes for stochastic optimization. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–9. IEEE, 2018.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations*, 2015.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[Kulunchakov and Mairal, 2019] Andrei Kulunchakov and Julien Mairal. A generic acceleration framework for stochastic composite optimization. In *Advances in Neural Information Processing Systems*, pages 12556–12567, 2019.

[LeCun and Cortes, 2010] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[Liu and Belkin, 2020] Chaoyue Liu and Mikhail Belkin. Accelerating SGD with momentum for over-parameterized learning. In *International Conference on Learning Representations*, 2020.

[Ma and Yarats, 2019] Jerry Ma and Denis Yarats. Quasi-hyperbolic momentum and Adam for deep learning. In *International Conference on Learning Representations*, 2019.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[Nesterov, 1983] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

[Nesterov, 2005] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103:127–152, 2005.

[Nesterov, 2013] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[Ochs *et al.*, 2014] Peter Ochs, Yunjin Chen, Thomas Brox, and Thomas Pock. ipiano: Inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.

[Ong, 2017] Ming Yang Ong. *Understanding generalization*. PhD thesis, Massachusetts Institute of Technology, 2017.

[Polyak, 1964] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[Rakhlin *et al.*, 2012] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *International Conference on Machine Learning*, pages 1571–1578, 2012.

[Ramezani-Kebrya *et al.*, 2024] Ali Ramezani-Kebrya, Kimon Antonakopoulos, Volkan Cevher, Ashish Khisti, and Ben Liang. On the generalization of stochastic gradient descent with momentum. *Journal of Machine Learning Research*, 25(22):1–56, 2024.

[Robbins and Monro, 1951] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[Stephan *et al.*, 2017] Mandt Stephan, Matthew D. Hoffman, and David M. Blei. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18(134):1–35, 2017.

[Sun *et al.*, 2019a] Tao Sun, Dongsheng Li, Zhe Quan, Hao Jiang, Shengguo Li, and Yong Dou. Heavy-ball algorithms always escape saddle points. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, pages 3520–3526. ijcai.org, 2019.

[Sun *et al.*, 2019b] Tao Sun, Yuejiao Sun, Dongsheng Li, and Qing Liao. General proximal incremental aggregated gradient algorithms: Better and novel results under general scheme. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[Sun *et al.*, 2019c] Tao Sun, Penghang Yin, Dongsheng Li, Chun Huang, Lei Guan, and Hao Jiang. Non-ergodic convergence analysis of heavy-ball algorithms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5033–5040, 2019.

[Sun *et al.*, 2020] Tao Sun, Linbo Qiao, and Dongsheng Li. Nonergodic complexity of proximal inertial gradient descents. *IEEE Transactions on Neural Networks and Learning Systems*, 32(10):4613–4626, 2020.

[Sun *et al.*, 2021] Tao Sun, Huaming Ling, Zuoqiang Shi, Dongsheng Li, and Bao Wang. Training deep neural networks with adaptive momentum inspired by the quadratic optimization. *arXiv preprint arXiv:2110.09057*, 2021.

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

[Vaswani *et al.*, 2019] Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1195–1204. PMLR, 2019.

[Wang *et al.*, 2022] Bao Wang, Tan M. Nguyen, Tao Sun, Andrea L. Bertozzi, Richard G. Baraniuk, and Stanley J. Osher. Scheduled restart momentum for accelerated stochastic gradient descent. *SIAM J. Imaging Sci.*, 15(2):738–761, 2022.

[Welling and Teh, 2011] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning*, pages 681–688. Citeseer, 2011.

[Wiegerinck *et al.*, 1994] Wim Wiegerinck, Andrzej Komoda, and Tom Heskes. Stochastic dynamics of learning with momentum in neural networks. *Journal of Physics A: Mathematical and General*, 27(13):4425, 1994.

[Yan *et al.*, 2018] Yan Yan, Tianbao Yang, Zhe Li, Qihang Lin, and Yi Yang. A unified analysis of stochastic momentum methods for deep learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 2955–2961, 7 2018.

[Yuan *et al.*, 2016] Kun Yuan, Bicheng Ying, and Ali H. Sayed. On the influence of momentum acceleration on online learning. *Journal of Machine Learning Research*, 17(192):1–66, 2016.