# Calibrating the Adaptive Learning Rate to Improve Convergence of ADAM

Qianqian Tong[1], Guannan Liang[1], Jinbo Bi[1,*]

*Computer Science and Engineering, University of Connecticut, Storrs, CT 06269*

**Abstract**

Adaptive gradient methods (AGMs) have become popular in optimizing the nonconvex problems in deep learning area. We revisit AGMs and identify that the adaptive learning rate (A-LR) used by AGMs varies significantly across the dimensions of the problem over epochs (i.e., anisotropic scale), which may lead to issues in convergence and generalization. All existing modified AGMs actually represent efforts in revising the A-LR. Theoretically, we provide a new way to analyze the convergence of AGMs and prove that the convergence rate of ADAM also depends on its hyper-parameter $\epsilon$, which has been overlooked previously. Based on these two facts, we propose a new AGM by calibrating the A-LR with an activation (*softplus*) function, resulting in the SADAM and SAMSGRAD methods. We further prove that these algorithms enjoy better convergence speed under nonconvex, non-strongly convex, and Polyak-Łojasiewicz conditions compared with ADAM. Empirical studies support our observation of the anisotropic A-LR and show that the proposed methods outperform existing AGMs and generalize even better than S-Momentum in multiple deep learning tasks.

*Keywords:* ADAM, Deep learning, Adaptive methods, Stochastic methods

## 1. Introduction

Many machine learning problems can be formulated as the minimization of an objective function $f$ of the form: $\min_{x\in\mathbb{R}^d} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$, where both $f$ and $f_i$ maybe nonconvex in deep learning. Stochastic gradient descent (SGD), its variants such as SGD with momentum (S-Momentum) [1, 2, 3, 4], and adaptive gradient methods (AGMs) [5, 6, 7] play important roles in deep learning area due to simplicity and wide applicability. In particular, AGMs often exhibit fast initial progress in training and are easy to implement in solving large scale

*Corresponding author

*Email addresses:* qianqian.tong@uconn.edu (Qianqian Tong), guannan.liang@uconn.edu (Guannan Liang), jinbo.bi@uconn.edu (Jinbo Bi)

optimization problems. The updating rule of AGMs can be generally written as:

$$x_{t+1} = x_t - \frac{\eta_t}{\sqrt{v_t}} \odot m_t, \tag{1}$$

where $\odot$ calculates element-wise product of the first-order momentum $m_t$ and the learning rate (LR) $\frac{\eta_t}{\sqrt{v_t}}$. There is fairly an agreement on how to compute $m_t$, which is a convex combination of previous $m_{t-1}$ and current stochastic gradient $g_t$, i.e., $m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t$, $\beta_1 \in [0,1]$. The LR consists of two parts: the base learning rate (B-LR) $\eta_t$ is a scalar which can be constant or decay over iterations. In our convergence analysis, we consider the B-LR as constant $\eta$. The adaptive learning rate (A-LR), $\frac{1}{\sqrt{v_t}}$, varies adaptively across dimensions of the problem, where $v_t \in \mathbb{R}^d$ is the second-order momentum calculated as a combination of previous and current squared stochastic gradients. Unlike the first-order momentum, the formula to estimate the second-order momentum varies in different AGMs. As the core technique in AGMs, A-LR opens a new regime of controlling LR, and allows the algorithm to move with different step sizes along the search direction at different coordinates.

The first known AGM is ADAGRAD [5] where the second-order momentum is estimated as $v_t = \sum_{i=1}^{t} g_i^2$. It works well in sparse settings, but the A-LR often decays rapidly for dense gradients. To tackle this issue, ADADELTA [7], RMSPROP [8], ADAM [6] have been proposed to use exponential moving averages of past squared gradients, i.e., $v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2$, $\beta_2 \in [0,1]$ and calculate the A-LR by $\frac{1}{\sqrt{v_t}+\epsilon}$ where $\epsilon > 0$ is used in case that $v_t$ vanishes to zero. In particular, ADAM has become the most popular optimizer in the deep learning area due to its effectiveness in early training stage. Nevertheless, it has been empirically shown that ADAM generalizes worse than S-Momentum to unseen data and leaves a clear generalization gap [9, 10, 11], and even fails to converge in some cases [12, 13]. AGMs decrease the objective value rapidly in early iterations, and then stay at a plateau whereas SGD and S-Momentum continue to show dips in the training error curves, and thus continue to improve test accuracy over iterations. It is essential to understand what happens to ADAM in the later learning process, so we can revise AGMs to enhance their generalization performance.

Recently, a few modified AGMs have been developed, such as, AMSGRAD [12], YOGI [14], and ADABOUND [13]. AMSGRAD is the first method to theoretically address the non-convergence issue of ADAM by taking the largest second-order momentum estimated in the past iterations, i.e., $v_t = max\{v_{t-1}, \tilde{v}_t\}$ where $\tilde{v}_t = \beta_2 \tilde{v}_{t-1} + (1-\beta_2)g_t^2$, and proves its convergence in the convex case. The analysis is later extended to other AGMs (such as RMSPROP and AMSGRAD) in nonconvex settings [15, 16, 17, 18]. In latest study, ADAM has been proved to converge to a first-order stationary point with appropriate parameter setting [19]. Specifically, appropriate choices of momentum parameter $\beta_1$ can directly address the non-convergence of ADAM. YOGI claims that the past $g_t^2$'s are forgotten in a fairly fast manner in ADAM and proposes $v_t = v_{t-1} - (1-\beta_2)sign(v_{t-1} - g_t^2)g_t^2$ to adjust the decay rate of the A-LR.

However, the parameter $\epsilon$ in the A-LR is adjusted to $10^{-3}$, instead of $10^{-8}$ in the default setting of ADAM, so $\epsilon$ dominates the A-LR in later iterations when $v_t$ becomes small and can be responsible for performance improvement. The hyper-parameter $\epsilon$ has rarely been discussed previously and our analysis shows that the convergence rate is closely related to $\epsilon$, which is further verified in our experiments. PADAM[1] [20, 15] claims that the A-LR in ADAM and AMSGRAD are "overadapted", and proposes to replace the A-LR updating formula by $1/((v_t)^p + \epsilon)$ where $p \in (0, 1/2]$. ADABOUND confines the LR to a predefined range by applying $Clip(\frac{\eta}{\sqrt{v_t}}, \eta_l, \eta_r)$, where LR values outside the interval $[\eta_l, \eta_r]$ are clipped to the interval edges. However, a more effective way is to softly and smoothly calibrate the A-LR rather than hard-thresholding the A-LR at all coordinates. Our main contributions are summarized as follows:

1. We study AGMs from a new perspective: the range of the A-LR. Through experimental studies, we find that the A-LR is always anisotropic. This anisotropy may lead the algorithm to focus on a few dimensions (those with large A-LR), which may exacerbate generalization performance. We analyze the existing modified AGMs to help explain how they close the generalization gap.

2. Theoretically, we are the first to include hyper-parameter $\epsilon$ into the convergence analysis and clearly show that the convergence rate is upper bounded by a $1/\epsilon^2$ term, verifying prior observations that $\epsilon$ affects performance of ADAM empirically. We provide a new approach to convergence analysis of AGMs under the nonconvex, non-strongly convex, or Polyak-Łojasiewicz (P-L) condition.

3. Based on the above two results, we propose to calibrate the A-LR using an activation function, particularly we implement the *softplus* function with a hyper-parameter $\beta$, which can be combined with any AGM. In this work, we combine it with ADAM and AMSGRAD to form the SADAM and SAMSGRAD methods.

4. We also provide comprehensive theoretical analyses of our proposed methods, which recover the same convergence rate as SGD in terms of the maximum iteration $T$ as $O(1/\sqrt{T})$. Empirical evaluations show that our methods obviously increase test accuracy, and outperform many AGMs and even S-Momentum in multiple deep learning models.

## 2. Preliminaries

**Notations.** For any vectors $a, b \in \mathbb{R}^d$, we use $a \odot b$ for element-wise product, $a^2$ for element-wise square, $\sqrt{a}$ for element-wise square root, $a/b$ for element-

[1]The PADAM in [20] actually used AMSGRAD, and for clear comparison, we named it PAMSGRAD. In our experiments, we also compared with the ADAM that used the A-LR formula with $p$, which we named PADAM.

wise division; we use $a^k$ to denote element-wise power of $k$, and $\|a\|$ to denote its $l_2$-norm. We use $\langle a, b\rangle$ to denote their inner product, $max\{a, b\}$ to compute element-wise maximum. $e$ is the Euler number, $\log(\cdot)$ denotes logarithm function with base $e$, and $O(\cdot)$ to hide constants which do not rely on the problem parameters.

**Optimization Terminology.** In convex setting, the optimality gap, $f(x_t) - f^*$, is examined where $x_t$ is the iterate at iteration $t$, and $f^*$ is the optimal value attained at $x^*$ assuming that $f$ does have a minimum. When $f(x_t) - f^* \leq \delta$, it is said that the method reaches an optimal solution with $\delta$-accuracy. However, in the study of AGMs, the average regret $\frac{1}{T}\sum_{t=1}^{T}(f(x_t) - f^*)$ (where the maximum iteration number $T$ is pre-specified) is used to approximate the optimality gap to define $\delta$-accuracy. Our analysis moves one step further to examine if $f(\frac{1}{T}\sum_{t=1}^{T} x_t) - f^* \leq \delta$ by applying Jensen's inequality to the regret.

In nonconvex setting, finding the global minimum or even local minimum is NP-hard, so optimality gap is not examined. Rather, it is common to evaluate if a first-order stationary point has been achieved [21, 12, 14]. More precisely, we evaluate if $E[\|\nabla f(x_t)\|^2] \leq \delta$ (e.g., in the analysis of SGD [1]). The convergence rate of SGD is $O(1/\sqrt{T})$ in both non-strongly convex and nonconvex settings. Requiring $O(1/\sqrt{T}) \leq \delta$ yields the maximum number of iterations $T = O(1/\delta^2)$. Thus, SGD can obtain a $\delta$-accurate solution in $O(1/\delta^2)$ steps in non-strongly convex and nonconvex settings. Our results recover the rate of SGD and S-Momentum in terms of $T$.

**Assumption 1.** *The loss functions $f_i$ and the objective $f$ satisfy:*

*(a)* ***L-smoothness.*** $\forall x, y \in \mathbb{R}^d, \forall i \in \{1, ..., n\}$, $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$.

*(b) The noisy gradient is unbiased and the noise is independent, i.e., $\forall x \in \mathbb{R}^d$, $t \geq 1$, $g_t = \nabla f(x_t) + \xi_t, E[\xi_t] = 0$ and $\xi_i$ is independent of $\xi_j$ if $i \neq j$.*

*(c) At time $t$, the algorithm can access a bounded noisy gradient and the true gradient is bounded, i.e., $\forall t \geq 1, \|\nabla f(x_t)\| \leq H, \|g_t\| \leq H, H \geq 0$.*

Assumptions (a,b) are widely used in stochastic optimization analyses. Reference [16] has proposed and used assumption (c) as a further development of bounded elements of the gradient assumptions used in [12]. Notice that the full gradient with bounded norm is equivalent to the Lipschitz continuous assumption of $f$ when $f$ is differentiable. This assumption can be satisfied in practice.

**Definition 1.** *Suppose $f$ has the global minimum, denoted as $f^* = f(x^*)$. Then for any $x, y \in \mathbb{R}^d$,*

1. ***Non-strongly convex.*** $f(y) \geq f(x) + \nabla f(x)^T(y - x)$.
2. ***Polyak-Łojasiewicz (P-L) condition.*** $\exists \lambda > 0$ *such that* $\|\nabla f(x)\|^2 \geq 2\lambda(f(x) - f^*)$.
3. ***Strongly convex.*** $\exists\ \mu > 0$ *such that* $f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{\mu}{2}\|y - x\|^2$.

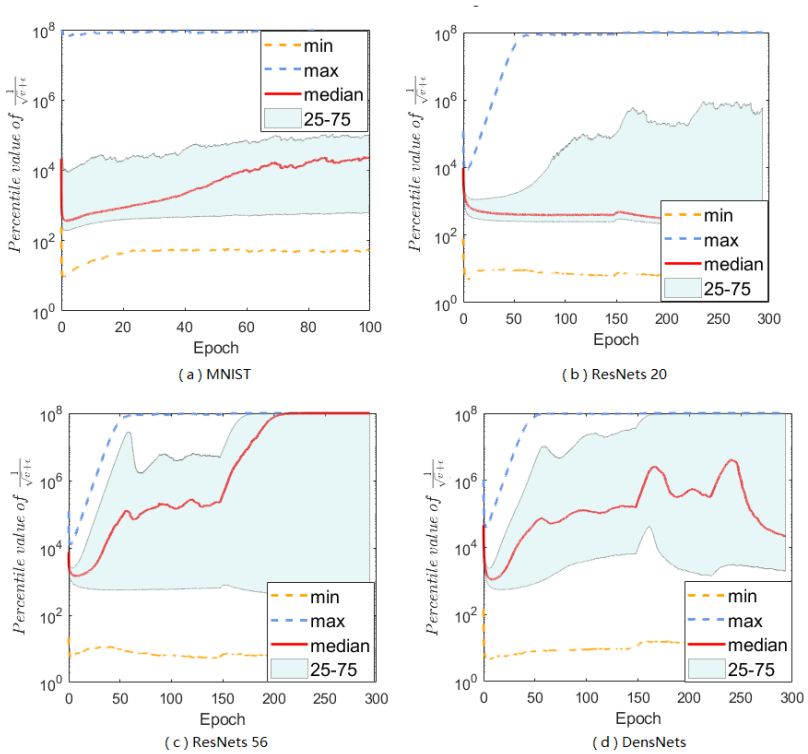


Figure 1: Range of the A-LR in ADAM over iterations in four settings: (a) CNN on MNIST, (b) ResNet20 on CIFAR-10, (d) ResNet56 on CIFAR-10, (d) DenseNets on CIFAR-10. We plot the min, max, median, and the 25 and 75 percentiles of the A-LR across dimensions (the elements in $\frac{1}{\sqrt{v_t}+\epsilon}$).

## 3. Our New Analysis of Adam

First, we empirically observe that ADAM has anisotropic A-LR caused by $\epsilon$, which may lead to poor generalization performance. Second,we theoretically show ADAM method is sensitive to $\epsilon$, supporting observations in previous work.

### *3.1. Anisotropic A-LR.*

We investigate how the A-LR in ADAM varies over time and across problem dimensions, and plot four examples in Figure 1 (more figures in Appendix) where we run ADAM to optimize a convolutional neural network (CNN) on the MNIST dataset, and ResNets or DenseNets on the CIFAR-10 dataset. The curves in Figure 1 exhibit very irregular shapes, and the median value is hardly placed in the middle of the range, the range of A-LR across the problem dimensions is anisotropic for AGMs. As a general trend, the A-LR becomes larger when $v_t$ approaches 0 over iterations. The elements in the A-LR vary significantly across

Table 1: Test Accuracy(%) of ADAM for different $\epsilon$.

| $\epsilon$ | ResNets 20 | ResNets 56 | DenseNets | ResNet 18 | VGG |
|---|---|---|---|---|---|
| $10^{-1}$ | $92.51 \pm 0.13$ | $94.29 \pm 0.10$ | $94.78 \pm 0.19$ | $77.21 \pm 0.26$ | $76.05 \pm 0.27$ |
| $10^{-2}$ | $92.88 \pm 0.21$ | $94.15 \pm 0.17$ | $94.35 \pm 0.10$ | $76.64 \pm 0.24$ | $75.69 \pm 0.16$ |
| $10^{-4}$ | $92.03 \pm 0.21$ | $93.62 \pm 0.18$ | $94.15 \pm 0.12$ | $76.19 \pm 0.20$ | $74.45 \pm 0.19$ |
| $10^{-6}$ | $92.99 \pm 0.22$ | $93.56 \pm 0.15$ | $94.24 \pm 0.24$ | $76.09 \pm 0.20$ | $74.20 \pm 0.33$ |
| $10^{-8}$ | $91.68 \pm 0.12$ | $92.82 \pm 0.09$ | $93.32 \pm 0.06$ | $76.14 \pm 0.24$ | $74.18 \pm 0.15$ |

Table 2: Test Accuracy(%) of AMSGrad for different $\epsilon$.

| $\epsilon$ | ResNets 20 | ResNets 56 | DenseNets | ResNet 18 | VGG |
|---|---|---|---|---|---|
| $10^{-1}$ | $92.80 \pm 0.22$ | $94.12 \pm 0.07$ | $94.92 \pm 0.10$ | $77.26 \pm 0.30$ | $75.84 \pm 0.16$ |
| $10^{-2}$ | $92.89 \pm 0.07$ | $94.20 \pm 0.18$ | $94.43 \pm 0.22$ | $76.23 \pm 0.26$ | $75.37 \pm 0.18$ |
| $10^{-4}$ | $91.85 \pm 0.10$ | $93.50 \pm 0.14$ | $94.02 \pm 0.18$ | $76.30 \pm 0.31$ | $74.44 \pm 0.16$ |
| $10^{-6}$ | $91.98 \pm 0.23$ | $93.54 \pm 0.16$ | $94.17 \pm 0.10$ | $76.14 \pm 0.16$ | $74.17 \pm 0.28$ |
| $10^{-8}$ | $91.70 \pm 0.12$ | $93.10 \pm 0.11$ | $93.71 \pm 0.05$ | $76.32 \pm 0.11$ | $74.26 \pm 0.18$ |

dimensions and there are always some coordinates in the A-LR of AGMs that reach the maximum $10^8$ determined by $\epsilon$ (because we use $\epsilon = 10^{-8}$ in ADAM).

This anisotropic scale of A-LR across dimensions makes it difficult to determine the B-LR, $\eta$. On the one hand, $\eta$ should be set small enough so that the LR $\frac{\eta}{\sqrt{v_t}+\epsilon}$ is appropriate, or otherwise some coordinates will have very large updates because the corresponding A-LR's are big, likely resulting in performance oscillation [22]. This may be due to that exponential moving average of past gradients is different, hence the speed of $m_t$ diminishing to zero is different from the speed of $\sqrt{v_t}$ diminishing to zero. Besides, noise generated in stochastic algorithms has nonnegligible influence to the learning process. On the other hand, very small $\eta$ may harm the later stage of the learning process since the small magnitude of $m_t$ multiplying with a small step size (at some coordinates) will be too small to escape sharp local minimal, which has been shown to lead to poor generalization [23, 24, 25]. Further, in many deep learning tasks, stage-wise policies are often taken to decay the LR after several epochs, thus making the LR even smaller. To address the dilemma, it is essential to control the A-LR, especially when stochastic gradients get close to 0.

By analyzing previous modified AGMs that aim to close the generalization gap, we find that all these works can be summarized into one technique: constraining the A-LR, $1/(\sqrt{v_t}+\epsilon)$, to a reasonable range. Based on the observation of anisotropic A-LR, we propose a more effective way to calibrate the A-LR according to an activation function rather than hard-thresholding the A-LR at all coordinates, empirically improve generalization performance with theoretical guarantees of optimization.

*3.2. Sensitive to $\epsilon$.*

As a hyper-parameter in AGMs, $\epsilon$ is originally introduced to avoid the zero denominator issue when $v_t$ goes to 0, and has never been studied in the convergence analysis of AGMs. However, it has been empirically observed that AGMs can be sensitive to the choice of $\epsilon$ in [17, 14]. As shown in Figure 1, a smaller $\epsilon = 10^{-8}$ leads to a wide span of the A-LR across the different dimensions, whereas a bigger $\epsilon = 10^{-3}$ as used in YOGI, reduces the span. To better learn the effect caused by sensitive $\epsilon$, we conduct experiments in multiple datasets and results are shown in Table 1 and 2. The setting of $\epsilon$ is the main force causing anisotropy, unsatisfied, there has no theoretical result explains the effect of $\epsilon$ on AGMs. Inspired by our observation, we believe that the current convergence analysis for ADAM is not complete if omitting $\epsilon$.

Most of the existing convergence analysis follows the line in [12] to first project the sequence of the iterates into a minimization problem as $x_{t+1} = x_t - \frac{\eta}{\sqrt{v_t}} m_t = \min_x \|v_t^{1/4}\big(x - (x_t - \frac{\eta}{\sqrt{v_t}} m_t)\big)\|$, and then examine if $||v_t^{1/4}(x_{t+1} - x^*)||$ decreases over iterations. Hence, $\epsilon$ is not discussed in this line of proof because it is not included in the step size. In our later convergence analysis section, we introduce an important lemma, bounded A-LR, and by using the bounds of the A-LR (specifically, the lower bound $\mu_1$ and upper bound $\mu_2$ both containing $\epsilon$ for ADAM), we give a new general framework of prove (details in Appendix) to show the convergence rate for reaching an $x$ that satisfies $E[\|\nabla f(x_t)\|^2] \leq \delta$ in the nonconvex setting. Then, we also derive the optimality gap from the stationary point in the convex and P-L settings (strongly convex). Notice that in the original analysis of ADAM [12], one can not guarantee that $v_t \leq v_{t-1}$ holds, hence, our following theoretical analyses are typically for AMSGRAD method and SAMSGRAD method.

**Theorem 3.1.** ***[Nonconvex]*** *Suppose $f(x)$ is a nonconvex function that satisfies Assumption 1. Let $\eta_t = \eta = O(\frac{1}{\sqrt{T}})$,* AMSGRAD *has*

$$\min_{t=1,\ldots,T} E[\|\nabla f(x_t)\|^2] \leq \frac{C_1}{\sqrt{T}} + \frac{C_2}{T} + \frac{C_3}{T\sqrt{T}},$$

*where $C_1 = \frac{1}{\mu_1}[f(x_1) - f^*] + (\frac{L^2\mu_2^2}{2\mu_1}(\frac{\beta_1}{1-\beta_1})^2 + \frac{\mu_2^2}{2\mu_1} + \frac{L\mu_2^2}{\mu_1})H^2, C_2 = \frac{\beta_1(\mu_2-\mu_1)dGH}{(1-\beta_1)\mu_1}, C_3 = \frac{L\beta_1^2 dH^2(\mu_2^2-\mu_1^2)}{(1-\beta_1)^2\mu_1}$, $\mu_1$ and $\mu_2$ are A-LR parameters that are defined in Lemma 5.1. Because $L, G, H, \beta_1$ are constants, we have*

$$\min_{t=1,\ldots,T} E[\|\nabla f(x_t)\|^2] \leq O(\frac{1}{\epsilon^2\sqrt{T}} + \frac{d}{\epsilon T} + \frac{d}{\epsilon^2 T\sqrt{T}}).$$

We show theoretical guarantee of ADAM under nonconvex setting, which recovers the same convergence rate as SGD in terms of the maximum iteration $T$ as $O(1/\sqrt{T})$. Different from $O(\log(T)/\sqrt{T})$ in [16], we choose to set stepsize as $\eta_t = \eta = O(\frac{1}{\sqrt{T}})$ here and in the following analyses, which is widely used and easy to implement.

Assumption $E[\|x_t - x^*\|] \leq D$, $\forall t$ is commonly used in the proofs of SGD or AGMs to guarantee that the iterates do not go too far away from the optimal solution in the stochastic process. For fair comparison, we follow the same assumption used in [6] and provide the following result.

**Theorem 3.2.** ***[Non-strongly Convex]*** *Suppose $f(x)$ is a convex function that satisfies Assumption 1. Assume that $\forall t$, $E[\|x_t - x^*\|] \leq D$, for any $m \neq n$, $E[\|x_m - x_n\|] \leq D_\infty$, let $\eta_t = \eta = O(\frac{1}{\sqrt{T}})$,* AMSGrad *has a convergence rate that satisfies $f(\bar{x}_t) - f^* \leq O(\frac{d}{\epsilon^2\sqrt{T}})$, where $\bar{x}_t = \frac{1}{T}\sum_{t=1}^{T} x_t$.*

The item dominating the convergence rate of Adam is $O(\frac{\tilde{C}}{\sqrt{T}})$, where $\tilde{C} = \frac{D^2}{2\mu_1} + \frac{\mu_2^2}{\mu_1}H^2 + \frac{\beta_1 dH^2}{2\mu_1(1-\beta_1)}(\mu_2^2 - \mu_1^2) + \frac{\beta_1 D^2}{\mu_1(1-\beta_1)} + \frac{\beta_1^3 D_\infty^2}{\mu_1(1-\beta_1)^3} + \frac{\beta_1\mu_2^2}{(1-\beta_1)\mu_1}H^2$. With fixed $L, G, H, \beta_1, D, D_\infty$, we have $\tilde{C} = O(\frac{d}{\epsilon^2})$, which contains $\epsilon$ and dimension $d$. When $\epsilon$ becomes bigger, the convergence rate is better. This observation supports the discussion in our paper.

**Theorem 3.3.** ***[P-L Condition]*** *Suppose $f(x)$ has P-L condition (with parameter $\lambda$) holds under convex case, satisfying Assumption 1. Let $\eta_t = \eta = O(\frac{1}{T^2})$,* AMSGrad *has the convergence rate: $E[f(x_{T+1}) - f^*] \leq (1 - \frac{2\lambda\mu_1}{T^2})^T E[f(x_1) - f^*] + O(\frac{1}{T})$,*

The constants contained in big $O$ include $L, G, H, \beta_1$ and A-LR parameters $\mu_1$, and $\mu_2$. The P-L condition is weaker than strongly convex, and for the strongly-convex case, we also have:

**Corollary 3.3.1.** ***[Strongly Convex]*** *Suppose $f(x)$ is $\mu$-strongly convex function that satisfies Assumption 1. Let $\eta_t = \eta = O(\frac{1}{T^2})$,* AMSGrad *has the convergence rate: $E[f(x_{T+1}) - f^*] \leq (1 - \frac{2\mu\mu_1}{T^2})^T E[f(x_1) - f^*] + O(\frac{1}{T})$*

This is the first time to theoretically include $\epsilon$ into analysis. As expected, the convergence rate of AMSGrad is highly related with $\epsilon$. A bigger $\epsilon$ will enjoy a better convergence rate since $\epsilon$ will dominate the A-LR and behaves like S-Momentum; A smaller $\epsilon$ will preserve stronger "adaptivity", we need to find a better way to control $\epsilon$.

## 4. The Proposed Algorithms

We propose to use activation functions to calibrate AGMs, and specifically focus on using *softplus* funciton on top of Adam and AMSGrad methods.

### 4.1. Activation Functions Help

Activation functions (such as sigmoid, ELU, tanh) transfer inputs to outputs are widely used in deep learning area. As a well-studied activation function, $softplus(x) = \frac{1}{\beta}\log(1+e^{\beta x})$ is known to keep large values unchanged (behaved like function $y = x$) while smoothing out small values (see Figure 2 (a)). The target magnitude to be smoothed out can be adjusted by a hyper-parameter $\beta \in \mathbb{R}$. In our new algorithms, we introduce $softplus(\sqrt{v_t}) = \frac{1}{\beta}\log(1+e^{\beta\cdot\sqrt{v_t}})$ to smoothly calibrate the A-LR. This calibration brings the following benefits: (1) constraining extreme large-valued A-LR in some coordinates (corresponding to the small-values in $v_t$) while keeping others untouched with appropriate $\beta$. For the undesirable large values in the A-LR, the *softplus* function condenses them smoothly instead of hard thresholding. For other coordinates, the A-LR largely remains unchanged; (2) removing the sensitive parameter $\epsilon$ because the *softplus* function can be lower-bounded by a nonzero number when used on non-negative variables, $softplus(\cdot) \geq \frac{1}{\beta}\log 2$.

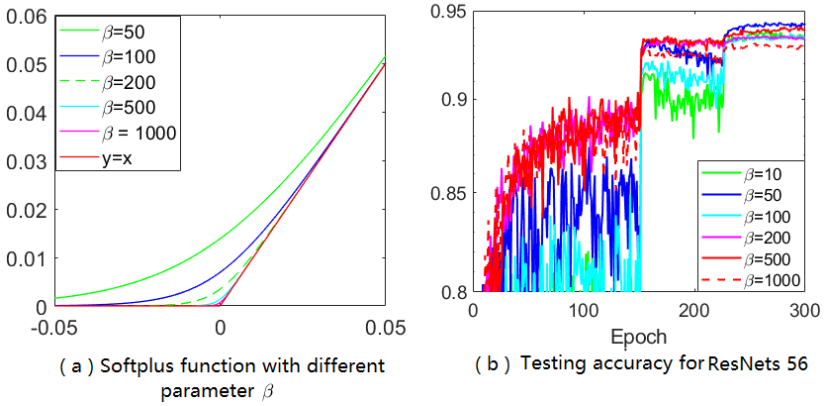


Figure 2: Behavior of the softplus function, and the test performance of our SADAM algorithm.

After calibrating $\sqrt{v_t}$ with a *softplus* function, the anisotropic A-LR becomes much more regulated (see Figure 3 and Appendix), and we clearly observe improved test accuracy (Figure 2 (b) and more figures in Appendix). We name this method "SADAM" to represent the calibrated ADAM with *softplus* function, here we recommend using *softplus* function but it is not limited to that, and the later theoretical analysis can be easily extended to other activation functions. More empirical evaluations have shown that the proposed methods significantly improve the generalization performance of ADAM and AMSGRAD.

### 4.2. Calibrated AGMs

With activation function, we develop two new variants of AGMs: SADAM and SAMSGRAD (Algorithms 1 and 2), which are developed based on ADAM and AMSGRAD respectively.

**Algorithm 1** SADAM

**Input:** $x_1 \in \mathbb{R}^d$, learning rate $\{\eta_t\}_{t=1}^T$, parameters $0 \leq \beta_1, \beta_2 < 1$, $\beta$.
**Initialize** $m_0 = 0$, $v_0 = 0$
**for** $t = 1$ to $T$ **do**
  Compute stochastic gradient $g_t$
  $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$
  $v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$
  $x_{t+1} = x_t - \frac{\eta_t}{softplus(\sqrt{v_t})} \odot m_t$
**end for**

**Algorithm 2** SAMSGRAD

**Input:** $x_1 \in \mathbb{R}^d$, learning rate $\{\eta_t\}_{t=1}^T$, parameters $0 \leq \beta_1, \beta_2 < 1$, $\beta$.
**Initialize** $m_0 = 0$, $\tilde{v}_0 = 0$
**for** $t = 1$ to $T$ **do**
  Compute stochastic gradient $g_t$
  $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$
  $\tilde{v}_t = \beta_2 \tilde{v}_{t-1} + (1-\beta_2) g_t^2$
  $v_t = max\{v_{t-1}, \tilde{v}_t\}$
  $x_{t+1} = x_t - \frac{\eta_t}{softplus(\sqrt{v_t})} \odot m_t$
**end for**

The key step lies in the way to design the adaptive functions, instead of using the generalized square root function only, we apply $softplus(\cdot)$ on top of the square root of the second-order momentum, which serves to regulate A-LR's anisotropic behavior and replace the tolerance parameter $\epsilon$ by the hyper-parameter $\beta$ used in the *softplus* function.

In our algorithms, the hyper-parameters are recommended as $\beta_1 = 0.9$, $\beta_2 = 0.999$. For clarity, we omit the bias correction step proposed in the original ADAM. However, our arguments and theoretical analysis are applicable to the bias correction version as well [6, 26, 14]. Using the *softplus* function, we introduce a new hyper-parameter $\beta$, which performs as a controller to smooth out anisotropic A-LR, and connect the ADAM and S-Momentum methods automatically. When $\beta$ is set to be small, SADAM and SAMSGRAD perform similarly to S-Momentum; when $\beta$ is set to be big, $softplus(\sqrt{v_t}) = \frac{1}{\beta}\log(1+e^{\beta\cdot\sqrt{v_t}}) \approx \frac{1}{\beta}\log(e^{\beta\cdot\sqrt{v_t}}) = \sqrt{v_t}$, and the updating formula becomes $x_{t+1} = x_t - \frac{\eta_t}{\sqrt{v_t}} \odot m_t$, which is degenerated into the original AGMs. The hyper-parameter $\beta$ can be well tuned to achieve the best performance for different datasets and tasks. Based on our empirical observations, we recommend to use $\beta = 50$.

As a calibration method, the *softplus* function has better adaptive behavior than simply setting $\epsilon$. More precisely, when $\epsilon$ is large or $\beta$ is small, ADAM and AMSGrad amount to S-Momentum, but when $\epsilon$ is small as commonly suggested $10^{-8}$ or $\beta$ is taken large, the two methods are different because comparing Figure 1 and 3 yields that SADAM has more regulated A-LR distribution. The proposed calibration scheme regulates the massive range of A-LR back down to a moderate scale. The median of A-LR in different dimensions is now well positioned to the middle of the 25-75 percentile zone. Our approach opens up a new direction to examine other activation functions (not limited to the *softplus* function) to calibrate the A-LR.

The proposed SADAM and SAMSGRAD can be treated as members of a class of AGMs that use the *softplus* (or another suitable activation) function to better adapt the step size. It can be readily combined with any other AGM,

e.g., RMSROP, YOGI, and PADAM. These methods may easily go back to the original ones by choosing a big $\beta$.

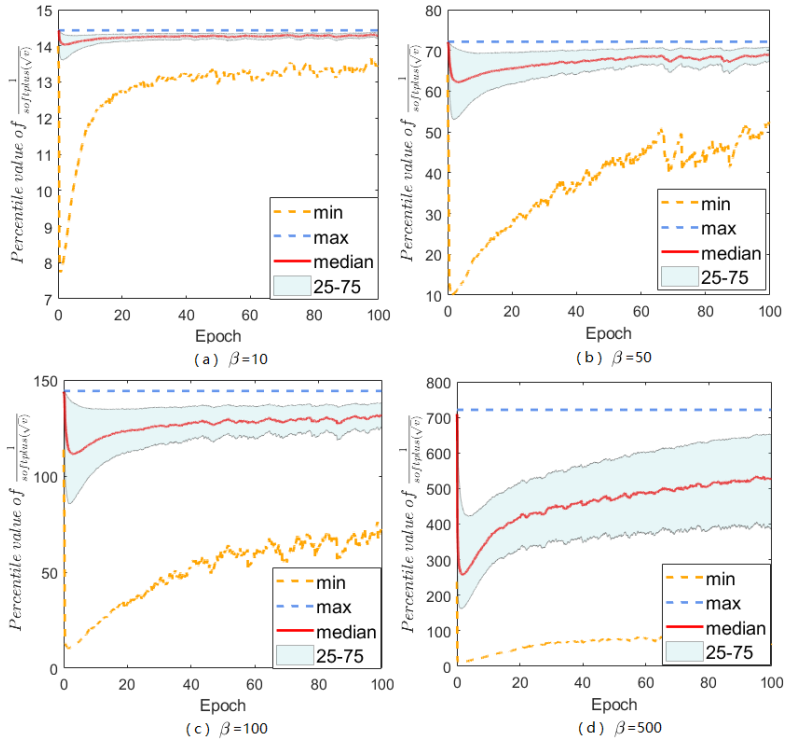


Figure 3: Behavior of the A-LR in the SADAM method with different choices of $\beta$ (CNN on the MNIST data).

## 5. Convergence Analysis

We first demonstrate an important lemma to highlight that every coordinate in the A-LR is both upper and lower bounded at all iterations, which is consistent with empirical observations (Figure 1), and forms the foundation of our proof.

**Lemma 5.1.** ***[Bounded A-LR]*** *With Assumption 1, for any $t \geq 1$, $j \in [1, d]$, $\beta_2 \in [0, 1]$, and $\epsilon$ in* ADAM*, $\beta$ in* SADAM*, anisotropic A-LR is bounded in AGMs,*
ADAM *has $(\mu_1, \mu_2)$-bounded A-LR:*

$$\mu_1 \leq \frac{1}{\sqrt{v_{t,j}} + \epsilon} \leq \mu_2,$$

SADAM *has $(\mu_3, \mu_4)$-bounded A-LR:*

$$\mu_3 \leq \frac{1}{softplus(\sqrt{v_{t,j}})} \leq \mu_4,$$

*where* $0 < \mu_1 \leq \mu_2$, *and* $0 < \mu_3 \leq \mu_4$.

**Remark 5.2.** *Besides the square root function and* softplus *function, the A-LR calibrated by any positive monotonically increasing function can be bounded. All of the bounds can be shown to be related to* $\epsilon$ *or* $\beta$ *(see Appendix). Bounded A-LR is an essential foundation in our analysis, we provide a different way of proof from previous works, and the proof procedure can be easily extended to other gradient methods as long as bounded LR is satisfied.*

**Remark 5.3.** *These bounds can be applied to all AGMs, including* ADAGRAD. *In fact, the lower bounds actually are not the same in* ADAM *and* ADAGRAD, *because* ADAM *will have smaller* $\sqrt{v_{t,j}}$ *due to moment decay parameter* $\beta_2$. *To achieve a unified result, we use the same relaxation to derive the fixed lower bound* $\mu_1$.

We now describe our main results of SAMSGRAD in the nonconvex case, we clearly show that similar to Theorem 3.1, the convergence rate of SAMSGRAD is related to the bounds of the A-LR. Our methods have improved the convergence rate of AMSGRAD when comparing self-contained parameters $\epsilon$ and $\beta$.

**Theorem 5.4.** ***[Nonconvex]*** *Suppose* $f(x)$ *is a nonconvex function that satisfies Assumption 1. Let* $\eta_t = \eta = O(\frac{1}{\sqrt{T}})$, SAMSGRAD *method has*

$$\min_{t=1,\ldots,T} E[\|\nabla f(x_t)\|^2] = \frac{C_1}{\sqrt{T}} + \frac{C_2}{T} + \frac{C_3}{T\sqrt{T}}$$

*where* $C_1 = \frac{1}{\mu_3}[f(x_1) - f^*] + (\frac{L^2\mu_4^2}{2\mu_3}(\frac{\beta_1}{1-\beta_1})^2 + \frac{\mu_4^2}{2\mu_3} + \frac{L\mu_4^2}{\mu_3})H^2, C_2 = \frac{\beta_1(\mu_4-\mu_3)dGH}{(1-\beta_1)\mu_3}$, *and* $C_3 = \frac{L\beta_1^2 d(\mu_4^2-\mu_3^2)}{(1-\beta_1)^2\mu_3}H^2$. *With fixed* $L, G, H, \beta_1$, *we have* $C_1 = O(\beta^2)$, $C_2 = O(d\beta)$, $C_3 = O(d\beta^2)$. *Therefore,*

$$\min_{t=1,\ldots,T} E[\|\nabla f(x_t)\|^2] \leq O(\frac{\beta^2}{\sqrt{T}} + \frac{d\beta}{T} + \frac{d\beta^2}{T\sqrt{T}}).$$

**Remark 5.5.** *Compared with the rate in Theorem 3.1, the convergence rate of* SAMSGRAD *relies on* $\beta$, *which can be a much smaller number* ($\beta = 50$ *as recommended) than* $\frac{1}{\epsilon}$ *(commonly* $\epsilon = 10^{-8}$ *in AGMs), showing that our methods have a better convergence rate than* AMSGRAD. *When* $\beta$ *is huge,* SAMSGRAD*'s rate is comparable to the classic* AMSGRAD. *When* $\beta$ *is small, the convergence rate will be* $O(\frac{1}{\sqrt{T}})$ *which recovers that of SGD [1].*

**Corollary 5.5.1.** *Treat* $\epsilon$ *or* $\beta$ *as a constant, then* SAMSGRAD *method with fixed* $L, \sigma, G, \beta_1$, *and* $\eta = O(\frac{1}{\sqrt{T}})$, *have complexity of* $O(\frac{1}{\sqrt{T}})$, *and thus call for* $O(\frac{1}{\delta^2})$ *iterations to achieve* $\delta$*-accurate solutions.*

**Theorem 5.6.** *[Non-strongly Convex] Suppose $f(x)$ is a convex function that satisfies Assumption 1. Assume that $E[\|x_t - x^*\|] \leq D$, $\forall t$, and $E[\|x_m - x_n\|] \leq D_\infty$, $\forall\ m \neq n$, let $\eta_t = \eta = O(\frac{1}{\sqrt{T}})$, SAMSGRAD has $f(\bar{x}_t) - f^* \leq O(\frac{1}{\sqrt{T}})$, where $\bar{x}_t = \frac{1}{T}\sum_{t=1}^{T} x_t$.*

The dominating item of convergence order of SADAM should be $O(\frac{\tilde{C}}{\sqrt{T}})$, where $\tilde{C} = \frac{D^2}{2\mu_3} + \frac{\mu_4^2 d}{\mu_3} H^2 + \frac{\beta_1 d H^2}{2\mu_3(1-\beta_1)}(\mu_4^2 - \mu_3^2) + \frac{\beta_1 D^2}{\mu_3(1-\beta_1)} + \frac{\beta_1^3 D_\infty^2}{\mu_3(1-\beta_1)^3} + \frac{\beta_1 \mu_4^2}{(1-\beta_1)\mu_3} H^2$. The accurate convergence rate will be $O(\frac{d}{\epsilon^2\sqrt{T}})$ for AMSGRAD and $O(\frac{d\beta^2}{\sqrt{T}})$ for SAMSGRAD with fixed $L$, $\sigma$, $G$, $\beta_1$, $D$, $D_\infty$. Some works may specify additional sparsity assumptions on stochastic gradients, and in other words, require that $\sum_{t=1}^{T}\sum_{j=1}^{d} \|g_{t,j}\| \ll \sqrt{dT}$ [5, 12, 15, 20] to reduce the order from $d$ to $\sqrt{d}$. Some works may use the element-wise bounds $\sigma_j$ or $G_j$, and apply $\sum_{j=1}^{d} \sigma_j = \sigma$, and $\sum_{j=1}^{d} G_j = G$ to hide $d$. In our work, we do not assume sparsity, so we use $\sigma$ and $G$ throughout the proof. Otherwise, those techniques can also be used to hide $d$ from our convergence rate.

**Corollary 5.6.1.** *If $\epsilon$ or $\beta$ is treated as constants, then SAMSGRAD method with fixed $L, \sigma, G, \beta_1$, and $\eta = O(\frac{1}{\sqrt{T}})$ in the convex case will call for $O(\frac{1}{\delta^2})$ iterations to achieve $\delta$-accurate solutions.*

**Theorem 5.7.** *[P-L Condition] Suppose $f(x)$ satisfies the P-L condition (with parameter $\lambda$) and Assumption 1 in the convex case. Let $\eta_t = \eta = O(\frac{1}{T^2})$, SAMSGRAD has:*

$$E[f(x_{T+1}) - f^*] \leq (1 - \frac{2\lambda\mu_3}{T^2})^T E[f(x_1) - f^*] + O(\frac{1}{T}).$$

The constants contained in big $O$ include $L, G, H, \beta_1$, and A-LR parameters $\mu_1$, and $\mu_2$. We also include the P-L condition situation as below.

**Corollary 5.7.1.** *[Strongly Convex] Suppose $f(x)$ is $\mu$-strongly convex function that satisfies Assumption 1. Let $\eta_t = \eta = O(\frac{1}{T^2})$, SAMSGRAD has the convergence rate:*

$$E[f(x_{T+1}) - f^*] \leq (1 - \frac{2\mu\mu_3}{T^2})^T E[f(x_1) - f^*] + O(\frac{1}{T}).$$

In summary, our methods share the same convergence rate as AMSGRAD, and enjoy even better convergence speed if comparing the common values chosen for the parameters $\epsilon$ and $\beta$. Our convergence rate recovers that of SGD and S-Momentum in terms of $T$ for a small $\beta$.

## 6. Experiments

We compare SADAM and SAMSGRAD against several state-of-the-art optimizers including S-Momentum, ADAM, AMSGRAD, YOGI, PADAM, PAMSGRAD, ADABOUND, and AMSBOUND. More results and architecture details are in Appendix.

**Experimental Setup.** We use three datasets for image classifications: MNIST, CIFAR-10 and CIFAR-100 and two datasets for LSTM language models: Penn Treebank dataset (PTB) and the WikiText-2 (WT2) dataset. The MNIST dataset is tested on a CNN with 5 hidden layers. The CIFAR-10 dataset is tested on Residual Neural Network with 20 layers (ResNets 20) and 56 layers (ResNets 56) [9], and DenseNets with 40 layers [11]. The CIFAR-100 dataset is tested on VGGNet [27] and Residual Neural Network with 18 layers (ResNets 18) [9]. The Penn Treebank dataset (PTB) and the WikiText-2 (WT2) dataset are tested on 3-layer LSTM models [28].

We train CNN on the MNIST data for 100 epochs, ResNets/DenseNets on CIFAR-10 for 300 epochs, with a weight decay factor of $5 \times 10^{-4}$ and a batch size of 128, VGGNet/ResNets on CIFAR-100 for 300 epochs, with a weight decay factor of 0.025 and a batch size of 128 and LSTM language models on 200 epochs. For the CIFAR tasks, we use a fixed multi-stage LR decaying scheme: the B-LR decays by 0.1 at the 150-th epoch and 225-th epoch, which is a popular decaying scheme used in many works [29, 18]. For the language tasks, we use a fixed multi-stage LR decaying scheme: the B-LR decays by 0.1 at the 100-th epoch and 150-th epoch. All algorithms perform grid search for hyper-parameters to choose from $\{10, 1, 0.1, 0.01, 0.001, 0.0001\}$ for B-LR, $\{0.9, 0.99\}$ for $\beta_1$ and $\{0.99, 0.999\}$ for $\beta_2$. For algorithm-specific hyper-parameters, they are tuned around the recommended values, such as $p \in \{\frac{1}{8}, \frac{1}{16}\}$ in PADAM and PAMSGRAD. For our algorithms, $\beta$ is selected from $\{10, 50, 100\}$ in SADAM and SAMSGRAD, though we do observe fine-tuning $\beta$ can achieve better test accuracy most of time. All experiments on CIFAR tasks are repeated for 6 times to obtain the mean and standard deviation for each algorithm.

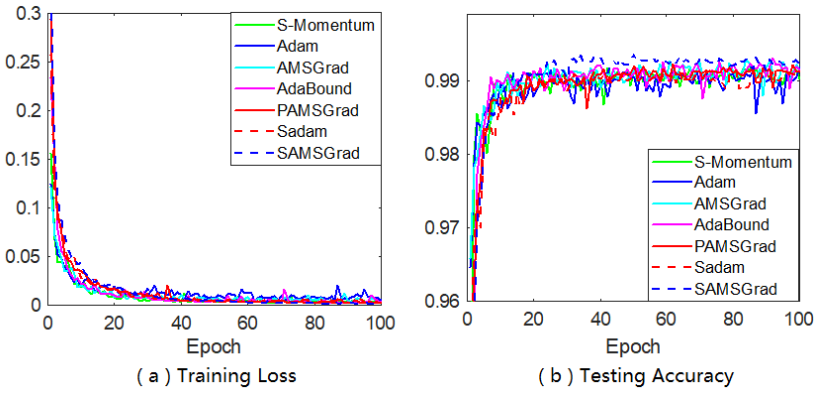


Figure 4: Training loss and test accuracy on MNIST.

Table 3: Test Accuracy(%) of CIFAR-10 for ResNets 20, ResNets 56 and DenseNets.

| Method | B-LR | $\epsilon$ | $\beta$ | ResNets 20 | ResNets 56 | DenseNets |
|---|---|---|---|---|---|---|
| S-Momentum [9, 11] | - | - | - | 91.25 | 93.03 | 94.76 |
| ADAM [14] | $10^{-3}$ | $10^{-3}$ | - | $92.56 \pm 0.14$ | $93.42 \pm 0.16$ | $93.35 \pm 0.21$ |
| YOGI [14] | $10^{-2}$ | $10^{-3}$ | - | $92.62 \pm 0.17$ | $93.90 \pm 0.21$ | $94.38 \pm 0.26$ |
| S-Momentum | $10^{-1}$ | - | - | $92.73 \pm 0.05$ | $94.11 \pm 0.15$ | $95.03 \pm 0.15$ |
| ADAM | $10^{-3}$ | $10^{-8}$ | - | $91.68 \pm 0.12$ | $92.82 \pm 0.09$ | $93.32 \pm 0.06$ |
| AMSGRAD | $10^{-3}$ | $10^{-8}$ | - | $91.7 \pm 0.12$ | $93.10 \pm 0.11$ | $93.71 \pm 0.05$ |
| PADAM | $10^{-1}$ | $10^{-8}$ | - | $92.7 \pm 0.10$ | $94.12 \pm 0.12$ | $95.06 \pm 0.06$ |
| PAMSGRAD | $10^{-1}$ | $10^{-8}$ | - | $92.74 \pm 0.12$ | $94.18 \pm 0.06$ | $\mathbf{95.21} \pm 0.10$ |
| ADABOUND | $10^{-2}$ | $10^{-8}$ | - | $91.59 \pm 0.24$ | $93.09 \pm 0.14$ | $94.16 \pm 0.10$ |
| AMSBOUND | $10^{-2}$ | $10^{-8}$ | - | $91.76 \pm 0.16$ | $93.08 \pm 0.09$ | $94.03 \pm 0.11$ |
| ADAM$^+$ | $10^{-1}$ | 0.013 | - | $92.89 \pm 0.13$ | $92.24 \pm 0.10$ | $94.54 \pm 0.13$ |
| AMSGRAD$^+$ | $10^{-1}$ | 0.013 | - | $\mathbf{92.95} \pm 0.17$ | $\mathbf{94.32} \pm 0.10$ | $94.58 \pm 0.18$ |
| SADAM | $10^{-2}$ | - | 50 | $\mathbf{93.01} \pm 0.16$ | $94.26 \pm 0.10$ | $95.19 \pm 0.18$ |
| SAMSGRAD | $10^{-2}$ | - | 50 | $92.88 \pm 0.10$ | $\mathbf{94.32} \pm 0.18$ | $\mathbf{95.31} \pm 0.15$ |

**Image Classification Tasks.** As a sanity check, experiment on MNIST has been done and its results are in Figure 4, which shows the learning curve for all baseline algorithms and our algorithms on both training and test datasets. As expected, all methods can reach the zero loss quickly, while for test accuracy, our SAMSGRAD shows increase in test accuracy and outperforms competitors within 50 epochs.

Using the PyTorch framework, we first run the ResNets 20 model on CIFAR10 and results are shown in Table 3. The original ADAM and AMSGRAD have lower test accuracy in comparison with S-Momentum, leaving a clear generalization gap exactly same as what is previously reported. For our methods, SADAM and SAMSGRAD clearly close the gap, and SADAM achieves the best test accuracy among competitors. We further test all methods with CIFAR10 on ResNets 56 with greater network depth, and the overall performance of each algorithm has been improved. For the experiments with DenseNets, we use a DenseNet with 40 layers and a growth rate $k = 12$ without bottleneck, channel reduction, or dropout. The results are reported in the last column of Table 3, SAMSGRAD still achieves the best test performance, and the proposed two methods largely improve the performance of ADAM and AMSGRAD and close the gap with S-Momentum.

Furthermore, two popular CNN architectures: VGGNet [27] and ResNets18 [9] are tested on CIFAR-100 dataset to compare different algorithms. Results can be found in Figure 5 and repeated results are in Appendix. Our proposed methods again perform slightly better than S-Momentum in terms of test accuracy.

**LSTM Language Models.** Observing the significant improvements in deep neural networks for image classification tasks, we further conduct experiments on the language models with LSTM. For comparing the efficiency of our

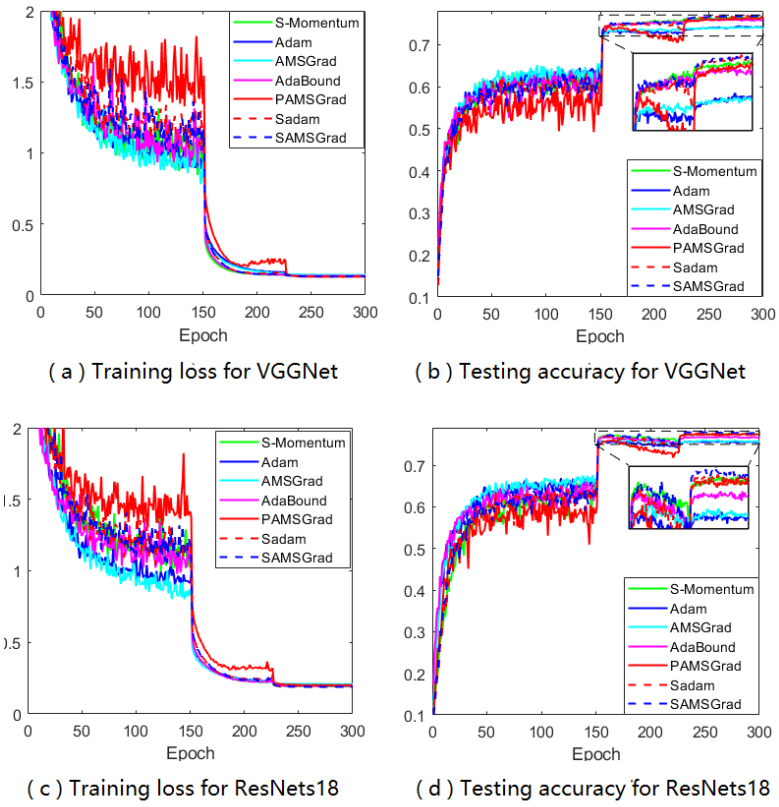


Figure 5: Training loss and test accuracy of two CNN architectures on CIFAR-100.

proposed methods, two LSTM models over the Penn Treebank dataset (PTB) [30] and the WikiText-2 (WT2) dataset [31] are tested. We present the single-model perplexity results for both our proposed methods and other competitive methods in Figure 6 and our methods achieve both fast convergence and best generalization performance.

In summary, our proposed methods show great efficacy on several standard benchmarks in both training and testing results, and outperform most optimizers in terms of generalization performance.

## 7. Conclusion

In this paper, we study adaptive gradient methods from a new perspective that is driven by the observation that the adaptive learning rates are anisotropic at each iteration. Inspired by this observation, we propose to calibrate the adaptive learning rates using an activation function, and in this work, we examine *softplus* function. We combine this calibration scheme with ADAM and AMSGRAD methods and empirical evaluations show obvious improvement on their

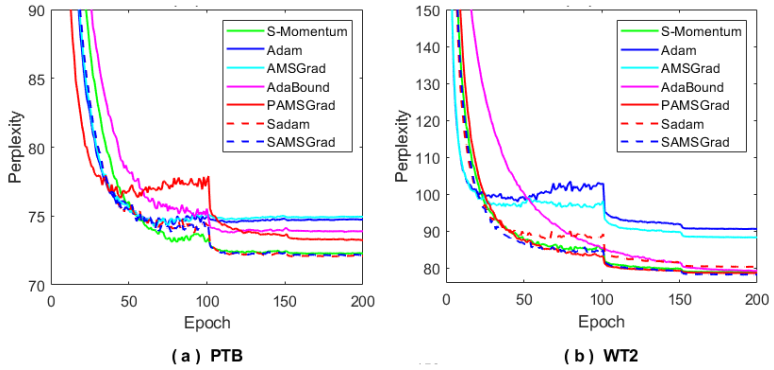


Figure 6: Perplexity curves on the test set on 3-layer LSTM models over PTB and WT2 datasets

generalization performance in multiple deep learning tasks. Using this calibration scheme, we replace the hyper-parameter $\epsilon$ in the original methods by a new parameter $\beta$ in the *softplus* function. A new mathematical model has been proposed to analyze the convergence of adaptive gradient methods. Our analysis shows that the convergence rate is related to $\epsilon$ or $\beta$, which has not been previously revealed, and the dependence on $\epsilon$ or $\beta$ helps us justify the advantage of the proposed methods. In the future, the calibration scheme can be designed based on other suitable activation functions, and used in conjunction with any other adaptive gradient method to improve generalization performance.

## Acknowledgments

This work was funded by NSF grants CCF-1514357, DBI-1356655, and IIS-1718738 to Jinbo Bi, who was also supported by NIH grants K02-DA043063 and R01-DA037349.

## References

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

[2] S. J. Wright, J. Nocedal, Numerical optimization, Springer Science 35 (67-68) (1999) 7.

[3] A. C. Wilson, B. Recht, M. I. Jordan, A lyapunov analysis of momentum methods in optimization, arXiv preprint arXiv:1611.02635.

[4] T. Yang, Q. Lin, Z. Li, Unified convergence analysis of stochastic momentum methods for convex and non-convex optimization, arXiv preprint arXiv:1604.03257.

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

[6] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

[7] M. D. Zeiler, Adadelta: an adaptive learning rate method, arXiv preprint arXiv:1212.5701.

[8] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, COURSERA: Neural networks for machine learning 4 (2) (2012) 26–31.

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

[10] S. Zagoruyko, N. Komodakis, Wide residual networks, arXiv preprint arXiv:1605.07146.

[11] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.

[12] S. J. Reddi, S. Kale, S. Kumar, On the convergence of adam and beyond.

[13] L. Luo, Y. Xiong, Y. Liu, X. Sun, Adaptive gradient methods with dynamic bound of learning rate, arXiv preprint arXiv:1902.09843.

[14] M. Zaheer, S. Reddi, D. Sachan, S. Kale, S. Kumar, Adaptive methods for nonconvex optimization, in: Advances in Neural Information Processing Systems, 2018, pp. 9815–9825.

[15] D. Zhou, Y. Tang, Z. Yang, Y. Cao, Q. Gu, On the convergence of adaptive gradient methods for nonconvex optimization, arXiv preprint arXiv:1808.05671.

[16] X. Chen, S. Liu, R. Sun, M. Hong, On the convergence of a class of adam-type algorithms for non-convex optimization, arXiv preprint arXiv:1808.02941.

[17] S. De, A. Mukherjee, E. Ullah, Convergence guarantees for rmsprop and adam in non-convex optimization and an empirical comparison to nesterov acceleration.

[18] M. Staib, S. J. Reddi, S. Kale, S. Kumar, S. Sra, Escaping saddle points with adaptive gradient methods, arXiv preprint arXiv:1901.09149.

[19] Z. Guo, Y. Xu, W. Yin, R. Jin, T. Yang, On stochastic moving-average estimators for non-convex optimization, arXiv preprint arXiv:2104.14840.

[20] J. Chen, Q. Gu, Closing the generalization gap of adaptive gradient methods in training deep neural networks, arXiv preprint arXiv:1806.06763.

[21] S. J. Reddi, A. Hefny, S. Sra, B. Poczos, A. J. Smola, On variance reduction in stochastic gradient descent and its asynchronous variants, in: Advances in Neural Information Processing Systems, 2015, pp. 2647–2655.

[22] R. Kleinberg, Y. Li, Y. Yuan, An alternative view: When does sgd escape local minima?, in: International Conference on Machine Learning, 2018, pp. 2703–2712.

[23] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, P. T. P. Tang, On large-batch training for deep learning: Generalization gap and sharp minima, arXiv preprint arXiv:1609.04836.

[24] P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, R. Zecchina, Entropy-sgd: Biasing gradient descent into wide valleys, arXiv preprint arXiv:1611.01838.

[25] H. Li, Z. Xu, G. Taylor, C. Studer, T. Goldstein, Visualizing the loss landscape of neural nets, in: Advances in Neural Information Processing Systems, 2018, pp. 6389–6399.

[26] T. Dozat, Incorporating nesterov momentum into adam.

[27] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[28] S. Merity, N. S. Keskar, R. Socher, Regularizing and Optimizing LSTM Language Models, arXiv preprint arXiv:1708.02182.

[29] N. S. Keskar, R. Socher, Improving generalization performance by switching from adam to sgd, arXiv preprint arXiv:1712.07628.

[30] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, S. Khudanpur, Recurrent neural network based language model, in: Eleventh annual conference of the international speech communication association, 2010.

[31] J. Bradbury, S. Merity, C. Xiong, R. Socher, Quasi-recurrent neural networks, arXiv preprint arXiv:1611.01576.

[32] M. Wang, E. X. Fang, H. Liu, Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions, Mathematical Programming 161 (1-2) (2017) 419–449.

## Appendix

### A.1. Architecture Used in Our Experiments

Here we mainly introduce the MNIST architecture with Pytorch used in our empirical study, ResNets and DenseNets are well-known architectures used in many works and we do not include details here.

| layer | layer setting |
|---|---|
| F.relu(self.conv1(x)) | self.conv1 = nn.Conv2d(1, 6, 5) |
| F.max_pool2d(x, 2, 2) | |
| F.relu(self.conv2(x)) | self.conv2 = nn.Conv2d(6, 16, 5) |
| x.view(-1, 16*4) | |
| F.relu(self.fc1(x)) | self.fc1 = nn.Linear(16*4*4, 120) |
| x= F.relu(self.fc2(x)) | self.fc2 = nn.Linear(120, 84) |
| x = self.fc3(x) | self.fc3 = nn.Linear(84, 10) |
| F.log_softmax(x, dim=1) | |

### B.2. More Empirical Results

In this section, we perform multiply experiments to study the property of anisotropic A-LR exsinting in AGMs and the performance of *softplus* function working on A-LR. We first show the A-LR range of popular Adam-type methods, then present how the parameter $\beta$ in Sadam and SAMSGrad reduce the range of A-LR and improve both training and testing performance.

#### *B.2.1. A-LR Range of AGMs*

Besides the A-LR range of Adam method, which has shown in main paper, we further want to study more other Adam-type methods, and do experiments focus on AMSGrad, PAdam, and PAMSGrad on different tasks (Figure B.2.1, B.2.2, and B.2.3). AMSGrad also has extreme large-valued coordinates, and will encounter the "small learning rate dilemma" as well as Adam. With partial parameter $p$, the value range of A-LR can be largely narrow down, and the maximum range will be reduced around $10^2$ with PAdam, and less than $10^2$ with PAMSGrad. This reduced range, avoiding the "small learning rate dilemma", may help us understand what "trick" works on Adam's A-LR can indeed improve the generalization performance. Besides, the range of A-LR in Yogi, AdaBound and AmsBound will be reduced or controlled by specific $\epsilon$ or *clip* function, we don't show more information here.

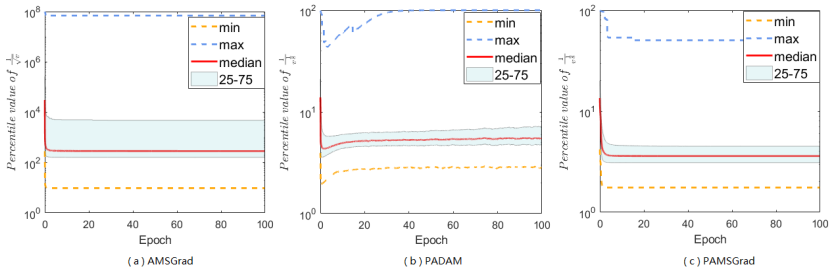


Figure B.2.1: A-LR range of AMSGrad (a), PAdam (b), and PAMSGrad (c) on MNIST.

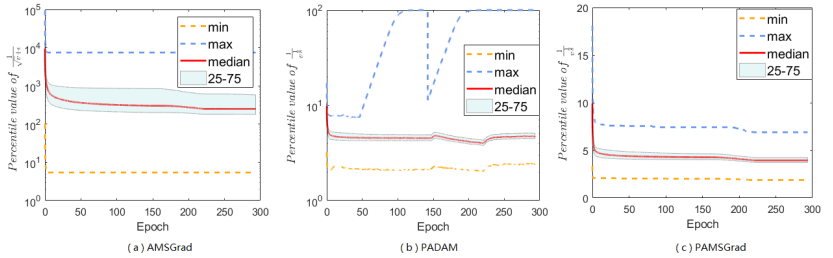


Figure B.2.2: A-LR range of AMSGrad (a), PAdam (b), and PAMSGrad (c) on ResNets 20.

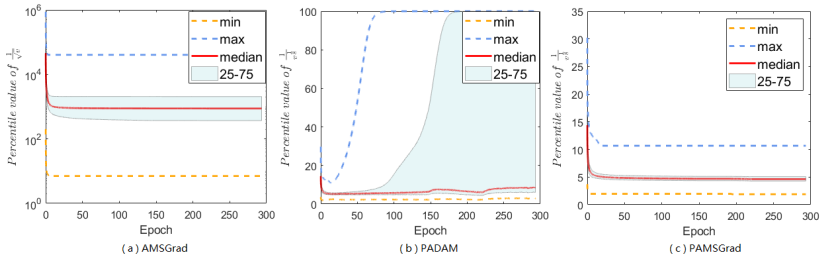


Figure B.2.3: A-LR range of AMSGrad (a), PAdam (b), and PAMSGrad (c) on DenseNets.

### *B.2.2. Parameter $\beta$ Reduces the Range of A-LR*

The main paper has discussed about *softplus* function, and mentions that it does help to constrain large-valued coordinates in A-LR while keep others untouched, here we give more empirical support. No matter how does $\beta$ set, the modified A-LR will have a reduced range. By setting various $\beta$'s, we can find an appropriate $\beta$ that performs the best for specific tasks on datasets. Besides the results of A-LR range of Sadam on MNIST with different choices of $\beta$, we also study Sadam and SAMSGrad on ResNets 20 and DenseNets.

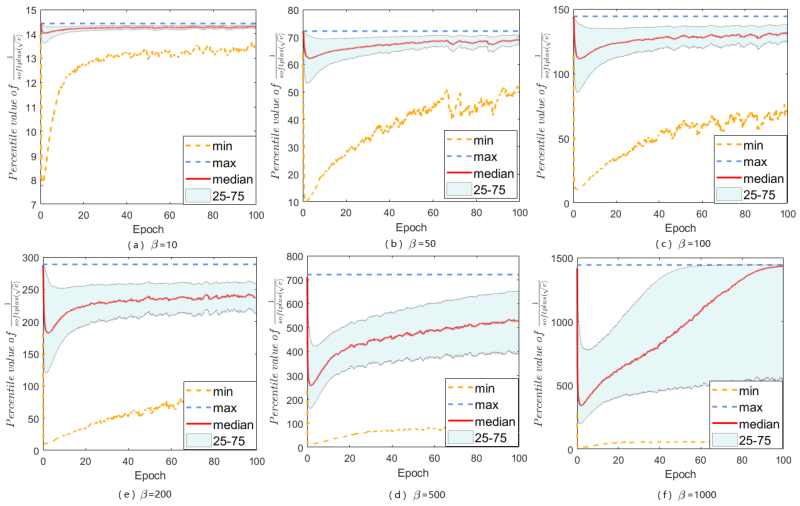


Figure B.2.4: The range of A-LR: $1/softplus(\sqrt{v_t})$ over iterations for SADAM on MNIST with different choices of $\beta$. The maximum ranges in all figures are compressed to a reasonable smaller value compared with $10^8$.

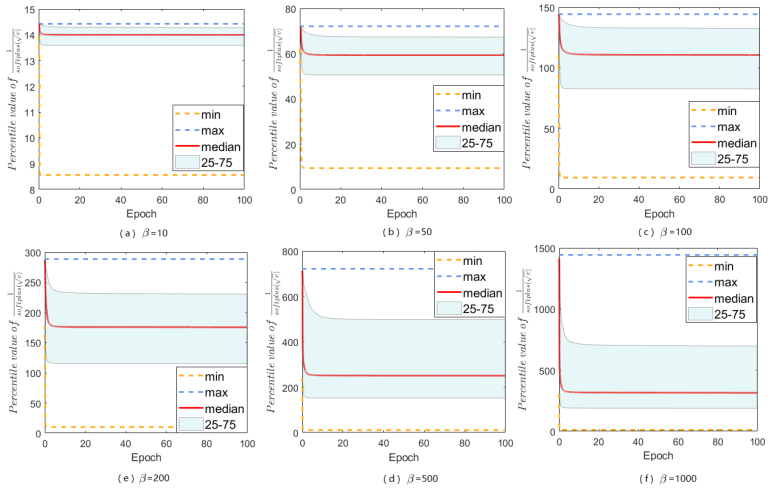


Figure B.2.5: The range of A-LR: $1/softplus(\sqrt{v_t})$, $v_t = max\{v_{t-1}, \tilde{v}_t\}$ over iterations for SAMSGRAD on MNIST with different choice of $\beta$. The maximum ranges in all figures are compressed to a reasonable smaller value compared with those of AMSGRAD on MNIST.

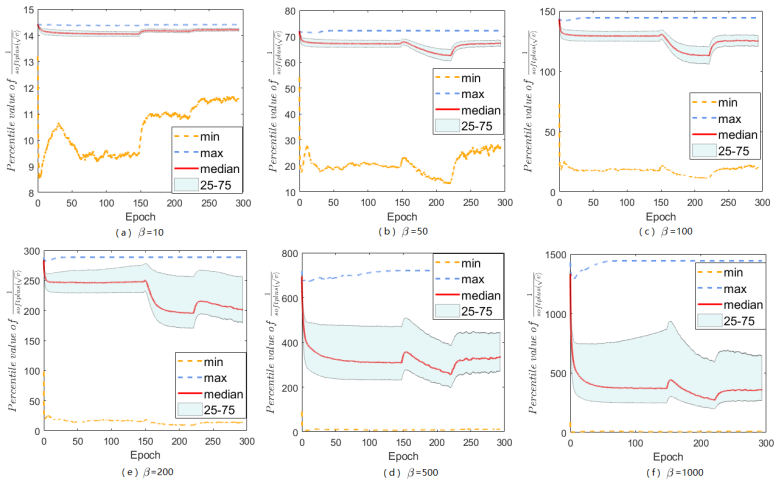


Figure B.2.6: The range of A-LR: $1/softplus(\sqrt{v_t})$ over iterations for SADAM on ResNets 20 with different choices of $\beta$.

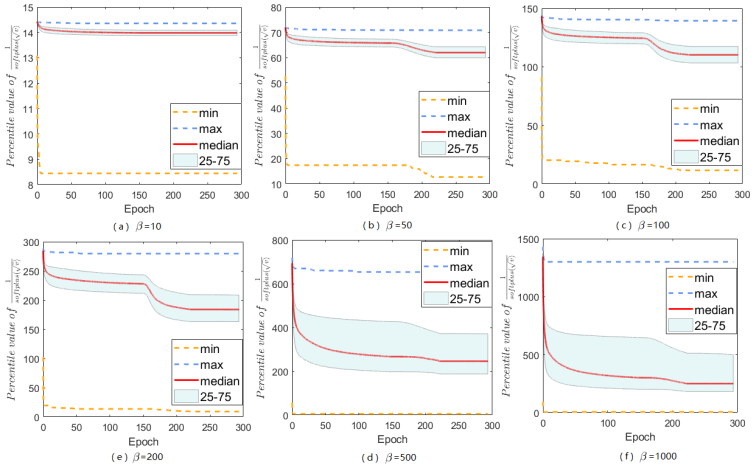


Figure B.2.7: The range of A-LR: $1/softplus(\sqrt{v_t})$, $v_t = max\{v_{t-1}, \tilde{v}_t\}$ over iterations for SAMSGRAD on ResNets 20 with different choices of $\beta$.

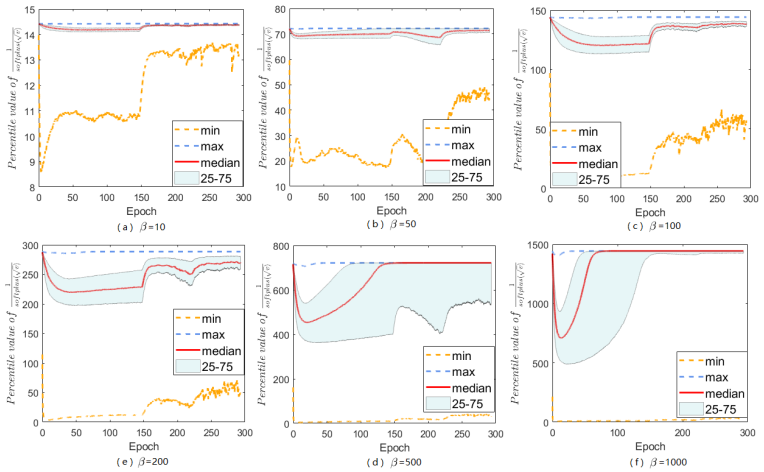


Figure B.2.8: The range of A-LR: $1/softplus(\sqrt{v_t})$ over iterations for SADAM on DenseNets with different choice of $\beta$.

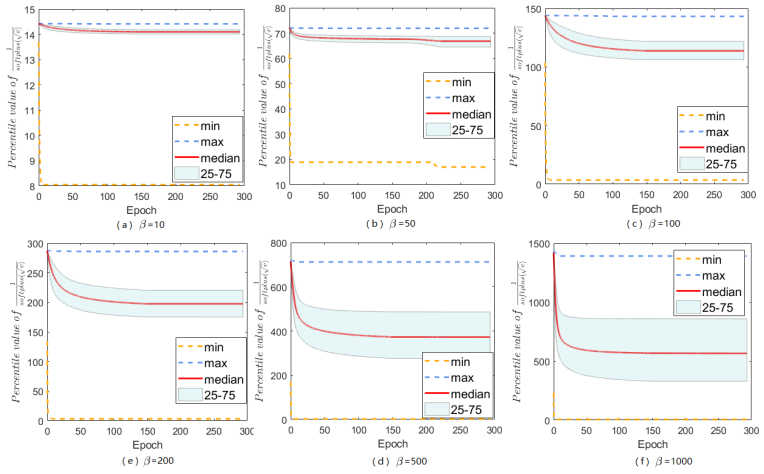


Figure B.2.9: The range of A-LR: $1/softplus(\sqrt{v_t})$, $v_t = max\{v_{t-1}, \tilde{v}_t\}$ over iterations for SAMSGRAD on DenseNets with different choices of $\beta$.

Here we do grid search to choose appropriate $\beta$ from $\{10, 50, 100, 200, 500, 1000\}$. In summary, with *softplus* fuction, SADAM and SAMSGRAD will narrow down the range of A-LR, make the A-LR vector more regular, avoiding "small learning rate dilemma" and finally achieve better performance.

*B.2.3. Parameter $\beta$ Matters in Both Training and Testing*

After studying existing ADAM-type methods, and effect of different $\beta$ in adjusting A-LR, we focus on the training and testing accuracy of our *softplus* framework, especially SADAM and SAMSGRAD, with different choices of $\beta$.

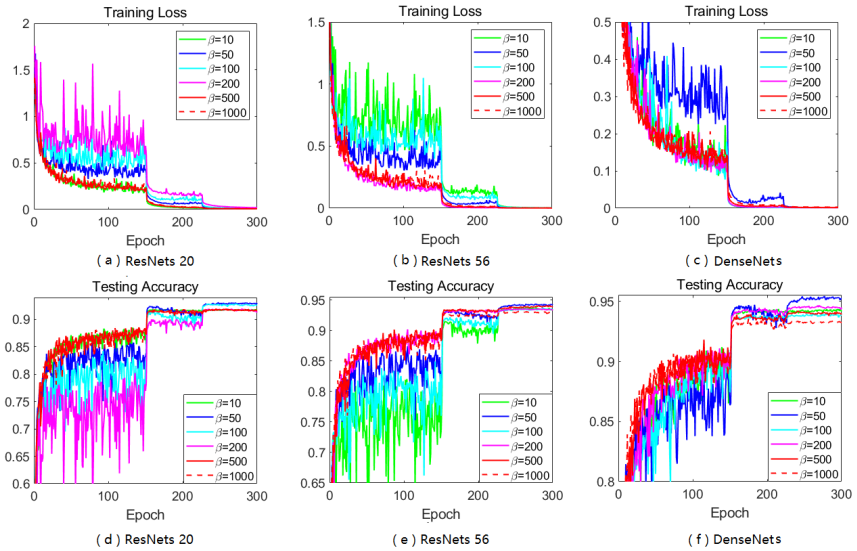


Figure B.2.10: Performance of SADAM on CIFAR-10 with different choice of $\beta$.

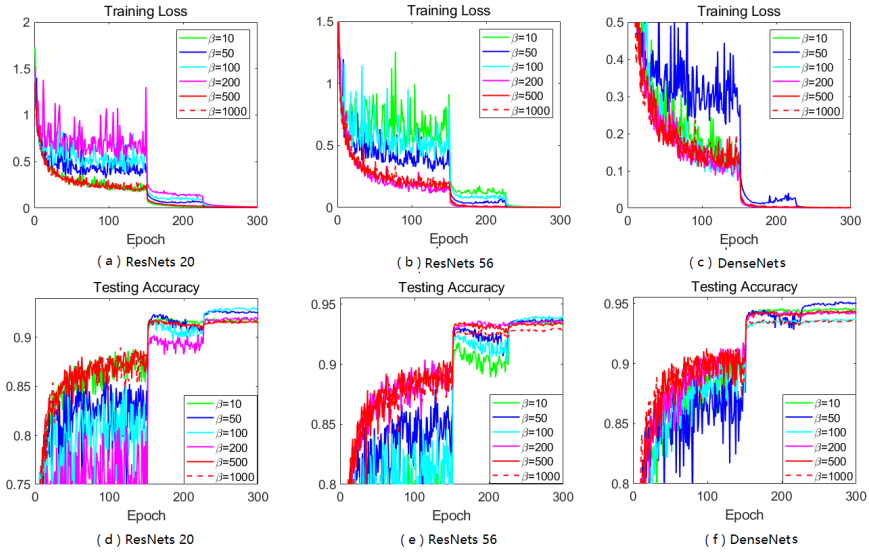


Figure B.2.11: Performance of SAMSGRAD on CIFAR-10 with different choice of $\beta$.

## C.3. CIFAR100

Two popular CNN architectures are tested on CIFAR-100 dataset to compare different algorithms: VGGNet [27] and ResNets18 [9]. Besides the figures in main text, we have repeated experiments and show results as follows. Our proposed methods again perform slightly better than S-Momentum in terms of test accuracy.

Table C.3.1: Test Accuracy(%) of CIFAR100 for VGGNet.

| Method | 50th epoch | 150th epoch | 250th epoch | best perfomance |
|---|---|---|---|---|
| S-Momentum | $59.09 \pm 2.09$ | $61.25 \pm 1.51$ | $76.14 \pm 0.12$ | $76.43 \pm 0.15$ |
| ADAM | $60.21 \pm 0.81$ | $62.98 \pm 0.10$ | $73.81 \pm 0.17$ | $74.18 \pm 0.15$ |
| AMSGRAD | $61.00 \pm 1.17$ | $63.27 \pm 1.18$ | $74.04 \pm 0.16$ | $74.26 \pm 0.18$ |
| PADAM | $53.62 \pm 1.70$ | $56.02 \pm 0.86$ | $75.85 \pm 0.20$ | $76.36 \pm 0.16$ |
| PAMSGRAD | $52.49 \pm 3.07$ | $57.39 \pm 1.40$ | $75.82 \pm 0.31$ | $76.26 \pm 0.30$ |
| ADABOUND | $60.27 \pm 0.99$ | $60.36 \pm 1.71$ | $75.86 \pm 0.23$ | $76.10 \pm 0.22$ |
| AMSBOUND | $59.88 \pm 0.56$ | $60.11 \pm 1.92$ | $75.74 \pm 0.23$ | $75.99 \pm 0.20$ |
| ADAM$^+$ | $43.59 \pm 2.71$ | $44.46 \pm 4.39$ | $74.91 \pm 0.36$ | $75.58 \pm 0.33$ |
| AMSGRAD$^+$ | $44.45 \pm 2.83$ | $45.61 \pm 3.67$ | $74.85 \pm 0.08$ | $75.56 \pm 0.24$ |
| SADAM | $58.59 \pm 1.60$ | $61.27 \pm 1.67$ | $76.35 \pm 0.18$ | $\mathbf{76.64} \pm 0.18$ |
| SAMSGRAD | $59.16 \pm 1.20$ | $60.86 \pm 0.39$ | $76.27 \pm 0.23$ | $76.47 \pm 0.26$ |

Table C.3.2: Test Accuracy(%) of CIFAR100 for ResNets18.

| Method | 50th epoch | 150th epoch | 250th epoch | best perfomance |
|---|---|---|---|---|
| S-Momentum | $59.98 \pm 1.31$ | $63.32 \pm 1.61$ | $77.19 \pm 0.36$ | $77.50 \pm 0.25$ |
| ADAM | $63.40 \pm 1.42$ | $66.18 \pm 1.02$ | $75.68 \pm 0.49$ | $76.14 \pm 0.24$ |
| AMSGRAD | $63.16 \pm 0.47$ | $66.59 \pm 1.42$ | $75.92 \pm 0.26$ | $76.32 \pm 0.11$ |
| PADAM | $56.28 \pm 0.87$ | $58.71 \pm 1.66$ | $77.18 \pm 0.21$ | $77.51 \pm 0.19$ |
| PAMSGRAD | $54.34 \pm 2.21$ | $58.81 \pm 1.95$ | $77.41 \pm 0.17$ | $77.67 \pm 0.14$ |
| ADABOUND | $61.13 \pm 0.84$ | $64.30 \pm 1.84$ | $77.18 \pm 0.38$ | $77.50 \pm 0.29$ |
| AMSBOUND | $61.05 \pm 1.59$ | $62.04 \pm 2.10$ | $77.08 \pm 0.19$ | $77.34 \pm 0.13$ |
| ADAM$^+$ | $46.5 \pm 2.12$ | $48.68 \pm 4.06$ | $76.86 \pm 0.36$ | $77.19 \pm 0.28$ |
| AMSGRAD$^+$ | $49.06 \pm 3.23$ | $50.75 \pm 2.45$ | $76.58 \pm 0.21$ | $76.91 \pm 0.12$ |
| SADAM | $59.00 \pm 1.09$ | $62.75 \pm 1.03$ | $77.26 \pm 0.30$ | $77.61 \pm 0.19$ |
| SAMSGRAD | $59.63 \pm 1.27$ | $63.44 \pm 1.84$ | $77.31 \pm 0.40$ | $\mathbf{77.70} \pm 0.31$ |

## D.4. Theoretical Analysis Details

We analyze the convergence rate of ADAM and SADAM under different cases, and derive competitive results of our methods. The following table gives an overview of stochastic gradient methods convergence rate under various conditions, in our work we provide a different way of proof compared with previous works and also associate the analysis with hyperparameters of ADAM methods.

*D.4.1. Prepared Lemmas*

We have a series of prepared lemmas to help with optimization convergence rate analysis, and some of them maybe also used in generalization error bound analysis.

**Lemma D.4.1.** *For any vectors $a, b, c \in \mathbb{R}^d$, $< a, b \odot c >=< a \odot b, c >=< a \odot \sqrt{b}, c \odot \sqrt{b} >$, here $\odot$ is element-wise product,$\sqrt{b}$ is element-wise square root.*

*Proof.*

$$< a, b \odot c > =< \begin{pmatrix} a_1 \\ \vdots \\ a_d \end{pmatrix}, \begin{pmatrix} b_1 c_1 \\ \vdots \\ b_d c_d \end{pmatrix} >= a_1 b_1 c_1 + \cdots + a_d b_d c_d$$

$$< a \odot b, c > =< \begin{pmatrix} a_1 b_1 \\ \vdots \\ a_d b_d \end{pmatrix}, \begin{pmatrix} c_1 \\ \vdots \\ c_d \end{pmatrix} >= a_1 b_1 c_1 + \cdots + a_d b_d c_d$$

$$< a \odot \sqrt{b}, c \odot \sqrt{b} > =< \begin{pmatrix} a_1 \sqrt{b_1} \\ \vdots \\ a_d \sqrt{b_d} \end{pmatrix}, \begin{pmatrix} \sqrt{b_1} c_1 \\ \vdots \\ \sqrt{b_d} c_d \end{pmatrix} >= a_1 b_1 c_1 + \cdots + a_d b_d c_d$$

□

**Lemma D.4.2.** *For any vector a, we have*

$$\|a^2\|_\infty \leq \|a\|^2. \tag{2}$$

**Lemma D.4.3.** *All momentum-based optimizers using first momentum $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$ will satisfy*

$$\|m_t\| \leq H. \tag{3}$$

*Proof.* We use induction to prove that $\|m_t\| \leq H$. Since $m_0 = 0$, we have $\|m_0\| \leq H$. Suppose $\|m_{t-1}\| \leq H$, we have $\|m_t\| = \|\beta_1 m_{t-1} + (1-\beta_1) g_t\| \leq (\beta_1 + (1-\beta_1)) max\{\|m_{t-1}\|, \|g_t\|\} = max\{\|m_{t-1}\|, \|g_t\|\} \leq H$. We can easily derive $\|m_t\|^2 \leq H^2$. Also, from the updating rule of first momentum estimator, we can derive

$$m_t = \Sigma_{i=1}^t (1-\beta_1) \beta_1^{t-i} g_i. \tag{4}$$

Let $\Gamma_t = \Sigma_{i=1}^t \beta_1^{t-i} = \frac{1-\beta_1^t}{1-\beta_1}$, by Jensen inequality and Assumption 1(c),

$$\begin{aligned} E[\|m_t\|^2] &= E[\|\Sigma_{i=1}^t (1-\beta_1)\beta_1^{t-i} g_i\|^2] = \Gamma_t^2 E[\|\Sigma_{i=1}^t \frac{(1-\beta_1)\beta_1^{t-i}}{\Gamma_t} g_i\|^2] \\ &\leq \Gamma_t^2 \Sigma_{i=1}^t (1-\beta_1)^2 \frac{\beta_1^{t-i}}{\Gamma_t} E[\|g_i\|^2] \leq \Gamma_t (1-\beta_1)^2 \Sigma_{i=1}^t \beta_1^{t-i} E[H^2] \\ &\leq H^2. \end{aligned}$$

□

**Lemma D.4.4.** *Each coordinate of vector $v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$ will satisfy*

$$E[v_{t,j}] \leq H^2,$$

*where $j \in [1, d]$ is the coordinate index.*

*Proof.* From the updating rule of second momentum estimator, we can derive

$$v_{t,j} = \Sigma_{i=1}^{t} (1-\beta_2)\beta_2^{t-i} g_{i,j}^2 \geq 0. \tag{5}$$

Since the decay parameter $\beta_2 \in [0, 1)$, $\Sigma_{i=1}^{t}(1-\beta_2)\beta_2^{t-i} = 1 - \beta_2^t \leq 1$. From Assumption 1(c),

$$E[v_{t,j}] = E[\Sigma_{i=1}^{t}(1-\beta_2)\beta_2^{t-i} g_{i,j}^2] \leq \Sigma_{i=1}^{t}(1-\beta_2)\beta_2^{t-i}(H^2) \leq H^2.$$

□

And we can derive the following important lemma:

**Lemma D.4.5.** ***[Bounded A-LR]*** *For any $t \geq 1$, $j \in [1, d]$, $\beta_2 \in [0, 1]$, and fixed $\epsilon$ in* AMSGrad *and $\beta$ defined in softplus function in* SAMSGrad*, the following bounds always hold:*

AMSGrad *has $(\mu_1, \mu_2)-$ bounded A-LR:*

$$\mu_1 \leq \frac{1}{\sqrt{v_{t,j}} + \epsilon} \leq \mu_2; \tag{6}$$

SAMSGrad *has $(\mu_3, \mu_4)-$ bounded A-LR:*

$$\mu_3 \leq \frac{1}{softplus(\sqrt{v_{t,j}})} \leq \mu_4; \tag{7}$$

*where $0 < \mu_1 \leq \mu_2$, $0 < \mu_3 \leq \mu_4$. For brevity, we use $\mu_l, \mu_u$ denoting the lower bound and upper bound respectively, and both* AMSGrad *and* SAMSGrad *will be analyzed with the help of $(\mu_l, \mu_u)$.*

*Proof.* For AMSGrad, let $\mu_1 = \frac{1}{H+\epsilon}$, $\mu_2 = \frac{1}{\epsilon}$, then we can get the result in (6).

For SAMSGrad, notice that $softplus(\cdot)$ is a monotone increasing function, and $\sqrt{v_{t,j}}$ is both upper-bounded and lower-bounded, then we have (7), where $\mu_3 = \frac{1}{\frac{1}{\beta}\log(1+e^{\beta \cdot H})}$, $\mu_4 = \frac{1}{\frac{1}{\beta}\log(1+e^{\beta \cdot 0})} = \frac{\beta}{\log 2}$. □

**Lemma D.4.6.** *Define* $z_t = x_t + \frac{\beta_1}{1-\beta_1}(x_t - x_{t-1}), \forall t \geq 1\ \beta_1 \in [0,1)$. *Let* $\eta_t = \eta$, *then the following updating formulas hold:*

*Gradient-based optimizer*

$$z_t = x_t, \quad z_{t+1} = z_t - \eta g_t; \tag{8}$$

AMSGRAD *optimizer*

$$z_{t+1} = z_t + \frac{\eta\beta_1}{1-\beta_1}(\frac{1}{\sqrt{v_{t-1}}+\epsilon} - \frac{1}{\sqrt{v_t}+\epsilon}) \odot m_{t-1} - \frac{\eta}{\sqrt{v_t}+\epsilon} \odot g_t; \tag{9}$$

SAMSGRAD *optimizer*

$$z_{t+1} = z_t + \frac{\eta\beta_1}{1-\beta_1}(\frac{1}{softplus(\sqrt{v_{t-1}})} - \frac{1}{softplus(\sqrt{v_t})}) \odot m_{t-1} - \frac{\eta}{softplus(\sqrt{v_t})} \odot g_t. \tag{10}$$

*Proof.* We consider the AMSGRAD optimizer and let $\beta_1 = 0$, we can easily derive the gradient-based case.

$$\begin{aligned}
z_{t+1} &= x_{t+1} + \frac{\beta_1}{1-\beta_1}(x_{t+1} - x_t) \\
z_{t+1} &= z_t + \frac{1}{1-\beta_1}(x_{t+1} - x_t) - \frac{\beta_1}{1-\beta_1}(x_t - x_{t-1}) \\
&= z_t - \frac{1}{1-\beta_1}\frac{\eta}{\sqrt{v_t}+\epsilon} \odot m_t + \frac{\beta_1}{1-\beta_1}\frac{\eta}{\sqrt{v_{t-1}}+\epsilon} \odot m_{t-1} \\
&= z_t + \frac{\eta\beta_1}{1-\beta_1}(\frac{1}{\sqrt{v_{t-1}}+\epsilon} - \frac{1}{\sqrt{v_t}+\epsilon}) \odot m_{t-1} - \frac{\eta}{\sqrt{v_t}+\epsilon} \odot g_t.
\end{aligned}$$

Similarly, consider the SAMSGRAD optimizer:

$$\begin{aligned}
z_{t+1} &= z_t + \frac{1}{1-\beta_1}(x_{t+1} - x_t) - \frac{\beta_1}{1-\beta_1}(x_t - x_{t-1}) \\
&= z_t - \frac{1}{1-\beta_1}\frac{\eta}{softplus(\sqrt{v_t})} \odot m_t + \frac{\beta_1}{1-\beta_1}\frac{\eta}{softplus(\sqrt{v_{t-1}})} \odot m_{t-1} \\
&= z_t + \frac{\eta\beta_1}{1-\beta_1}(\frac{1}{softplus(\sqrt{v_{t-1}})} - \frac{1}{softplus(\sqrt{v_t})}) \odot m_{t-1} - \frac{\eta}{softplus(\sqrt{v_t})} \odot g_t.
\end{aligned}$$

□

**Lemma D.4.7.** *As defined in Lemma D.4.6, and with the condition that* $v_t \geq v_{t-1}$, *we can derive the bound of distance of* $\|z_{t+1} - z_t\|^2$ *as follows:*
AMSGRAD *optimizer*

$$\begin{aligned}
E[\|z_{t+1} - z_t\|^2] \leq \frac{2\eta^2\beta_1^2 H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] \\
+ 2\eta^2\mu_2^2 H^2
\end{aligned} \tag{11}$$

SAMSGRAD *optimizer*

$$E[\|z_{t+1}-z_t\|^2] \leq \frac{2\eta^2\beta_1^2H^2}{(1-\beta_1)^2}E[\sum_{j=1}^{d}(\frac{1}{softplus(\sqrt{v_{t-1,j}})})^2-(\frac{1}{softplus(\sqrt{v_{t,j}})})^2] + 2\eta^2\mu_4^2H^2 \tag{12}$$

*Proof.* AMSGRAD case:

$$\begin{aligned}
E[\|z_{t+1}-z_t\|^2] &= E[\|\frac{\eta\beta_1}{1-\beta_1}(\frac{1}{\sqrt{v_{t-1}}+\epsilon}-\frac{1}{\sqrt{v_t}+\epsilon})\odot m_{t-1}-\frac{\eta}{\sqrt{v_t}+\epsilon}\odot g_t\|^2] \\
&\leq 2E[\|\frac{\eta\beta_1}{1-\beta_1}(\frac{1}{\sqrt{v_{t-1}}+\epsilon}-\frac{1}{\sqrt{v_t}+\epsilon})\odot m_{t-1}\|]^2+2E[\|\frac{\eta}{\sqrt{v_t}+\epsilon}\odot g_t\|]^2 \\
&\leq \frac{2\eta^2\beta_1^2H^2}{(1-\beta_1)^2}E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_{t-1,j}}+\epsilon}-\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2]+2\eta^2\mu_2^2H^2 \\
&\leq \frac{2\eta^2\beta_1^2H^2}{(1-\beta_1)^2}E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2-(\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2]+2\eta^2\mu_2^2H^2
\end{aligned}$$

The first inequality holds because $\|a-b\|^2 \leq 2\|a\|^2+2\|b\|^2$, the second inequality holds because Assumption 1(c) and D.4.3 and Lemma D.4.5, the third inequality holds because $(a-b)^2 \leq a^2-b^2$ when $a \geq b$, and in our assumption, we have $v_t \geq v_{t-1}$ holds.

SAMSGRAD case:

$$\begin{aligned}
E[\|z_{t+1}-z_t\|^2] &= E[\|\frac{\eta\beta_1}{1-\beta_1}(\frac{1}{softplus(\sqrt{v_{t-1}})}-\frac{1}{softplus(\sqrt{v_t})})\odot m_{t-1}-\frac{\eta}{softplus(\sqrt{v_t})}\odot g_t\|^2] \\
&\leq 2E[\|\frac{\eta\beta_1}{1-\beta_1}(\frac{1}{softplus(\sqrt{v_{t-1}})}-\frac{1}{softplus(\sqrt{v_t})})\odot m_{t-1}\|]^2 \\
&+ 2E[\|\frac{\eta}{softplus(\sqrt{v_t})}\odot g_t\|]^2 \\
&\leq \frac{2\eta^2\beta_1^2H^2}{(1-\beta_1)^2}E[\sum_{j=1}^{d}(\frac{1}{softplus(\sqrt{v_{t-1,j}})}-\frac{1}{softplus(\sqrt{v_{t,j}})})^2] \\
&+ 2\eta^2\mu_4^2H^2 \\
&\leq \frac{2\eta^2\beta_1^2H^2}{(1-\beta_1)^2}E[\sum_{j=1}^{d}(\frac{1}{softplus(\sqrt{v_{t-1,j}})})^2-(\frac{1}{softplus(\sqrt{v_{t,j}})})^2] \\
&+ 2\eta^2\mu_4^2H^2
\end{aligned}$$

Because the *softplus* function is monotone increasing function, therefore, the third inequality holds as well. □

**Lemma D.4.8.** *As defined in Lemma D.4.6, with the condition that $v_t \geq v_{t-1}$, we can derive the bound of the inner product as follows:*

AMSGRAD *optimizer*

$$-E[\langle \nabla f(z_t) - \nabla f(x_t), \frac{\eta}{\sqrt{v_t}+\epsilon} \odot g_t \rangle] \leq \frac{1}{2} L^2 \eta^2 \mu_2^2 (\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{1}{2} \eta^2 \mu_2^2 H^2; \quad (13)$$

SAMSGRAD *optimizer*

$$-E[\langle \nabla f(z_t) - \nabla f(x_t), \frac{\eta}{softplus(\sqrt{v_t})} \odot g_t \rangle] \leq \frac{1}{2} L^2 \eta^2 \mu_4^2 (\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{1}{2} \eta^2 \mu_4^2 H^2. \quad (14)$$

*Proof.* Since the stochastic gradient is unbiased, then we have $E[g_t] = \nabla f(x_t)$.

AMSGRAD case:

$$\begin{aligned}
&-E[\langle \nabla f(z_t) - \nabla f(x_t), \frac{\eta}{\sqrt{v_t}+\epsilon} \odot g_t \rangle] \\
&\leq \frac{1}{2} E[\|\nabla f(z_t) - \nabla f(x_t)\|^2] + \frac{1}{2} E[\|\frac{\eta}{\sqrt{v_t}+\epsilon} \odot g_t\|^2] \\
&\leq \frac{L^2}{2} E[\|z_t - x_t\|^2] + \frac{1}{2} E[\|\frac{\eta}{\sqrt{v_t}+\epsilon} \odot g_t\|^2] \\
&= \frac{L^2}{2} (\frac{\beta_1}{1-\beta_1})^2 E[\|x_t - x_{t-1}\|^2] + \frac{1}{2} E[\|\frac{\eta}{\sqrt{v_t}+\epsilon} \odot g_t\|^2] \\
&= \frac{L^2}{2} (\frac{\beta_1}{1-\beta_1})^2 E[\|\frac{\eta}{\sqrt{v_{t-1}}+\epsilon} \odot m_{t-1}\|^2] + \frac{1}{2} E[\|\frac{\eta}{\sqrt{v_t}+\epsilon} \odot g_t\|^2] \\
&\leq \frac{1}{2} L^2 \eta^2 \mu_2^2 (\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{1}{2} \eta^2 \mu_2^2 H^2
\end{aligned}$$

The first inequality holds because $\frac{1}{2}a^2 + \frac{1}{2}b^2 \geq - < a, b >$, the second inequality holds for L-smoothness, the last inequalities hold due to Lemma D.4.3 and D.4.5.

Similarly, for SAMSGRAD, we also have the following result:

$$
\begin{aligned}
&-E[\langle \nabla f(z_t) - \nabla f(x_t), \frac{\eta}{softplus(\sqrt{v_t})} \odot g_t \rangle] \\
&\quad \le \frac{1}{2} E[\|\nabla f(z_t) - \nabla f(x_t)\|^2] + \frac{1}{2} E[\| \frac{\eta}{softplus(\sqrt{v_t})} \odot g_t \|^2] \\
&\quad \le \frac{L^2}{2} E[\|z_t - x_t\|^2] + \frac{1}{2} E[\| \frac{\eta}{softplus(\sqrt{v_t})} \odot g_t \|^2] \\
&\quad = \frac{L^2}{2} (\frac{\beta_1}{1-\beta_1})^2 E[\|x_t - x_{t-1}\|^2] + \frac{1}{2} E[\| \frac{\eta}{softplus(\sqrt{v_t})} \odot g_t \|^2] \\
&\quad = \frac{L^2}{2} (\frac{\beta_1}{1-\beta_1})^2 E[\| \frac{\eta}{softplus(\sqrt{v_{t-1}})} \odot m_{t-1} \|^2] + \frac{1}{2} E[\| \frac{\eta}{softplus(\sqrt{v_t})} \odot g_t \|^2] \\
&\quad \le \frac{1}{2} L^2 \eta^2 \mu_4^2 (\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{1}{2} \eta^2 \mu_4^2 H^2.
\end{aligned}
$$

□

*D.4.2.* AMSGRAD *Convergence in Nonconvex Setting*

*Proof.* From L-smoothness and Lemma D.4.6, we have

$$
\begin{aligned}
f(z_{t+1}) &\le f(z_t) + \langle \nabla f(z_t), z_{t+1} - z_t \rangle + \frac{L}{2} \|z_{t+1} - z_t\|^2 \\
&= f(z_t) + \frac{\eta \beta_1}{1-\beta_1} \langle \nabla f(z_t), (\frac{1}{\sqrt{v_{t-1}} + \epsilon} - \frac{1}{\sqrt{v_t} + \epsilon}) \odot m_{t-1} \rangle \\
&\quad - \langle \nabla f(z_t), \frac{\eta}{\sqrt{v_t} + \epsilon} \odot g_t \rangle + \frac{L}{2} \|z_{t+1} - z_t\|^2
\end{aligned}
$$

Take the expectation on both sides produces

$$
\begin{aligned}
E[f(z_{t+1}) - f(z_t)] &\le \frac{\eta \beta_1}{1-\beta_1} E[\langle \nabla f(z_t), (\frac{1}{\sqrt{v_{t-1}} + \epsilon} - \frac{1}{\sqrt{v_t} + \epsilon}) \odot m_{t-1} \rangle] \\
&\quad - E[\langle \nabla f(z_t), \frac{\eta}{\sqrt{v_t} + \epsilon} \odot g_t \rangle] + \frac{L}{2} E[\|z_{t+1} - z_t\|^2] \\
&= \frac{\eta \beta_1}{1-\beta_1} E[\langle \nabla f(z_t), (\frac{1}{\sqrt{v_{t-1}} + \epsilon} - \frac{1}{\sqrt{v_t} + \epsilon}) \odot m_{t-1} \rangle] \\
&\quad - E[\langle \nabla f(z_t) - \nabla f(x_t), \frac{\eta}{\sqrt{v_t} + \epsilon} \odot g_t \rangle] - E[\langle \nabla f(x_t), \frac{\eta}{\sqrt{v_t} + \epsilon} \odot g_t \rangle] \\
&\quad + \frac{L}{2} E[\|z_{t+1} - z_t\|^2]
\end{aligned}
$$

Substituting the results of the lemmas yields,

$$E[f(z_{t+1}) - f(z_t)] \leq \frac{\eta\beta_1}{1-\beta_1} E[\langle \nabla f(z_t), (\frac{1}{\sqrt{v_{t-1}}+\epsilon} - \frac{1}{\sqrt{v_t}+\epsilon}) \odot m_{t-1}\rangle] + \frac{1}{2}L^2\eta^2\mu_2^2(\frac{\beta_1}{1-\beta_1})^2H^2 + \frac{1}{2}\eta^2\mu_2^2H^2 - E[\langle \nabla f(x_t), \frac{\eta}{\sqrt{v_t}+\epsilon} \odot g_t\rangle] + \frac{L\eta^2\beta_1^2H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] + L\eta^2\mu_2^2H^2$$

For the first term in the right hand side above, since $\|m_t\| \leq H$, we can further derive:

$$\begin{aligned}
&\frac{\eta\beta_1}{1-\beta_1} E[\langle \nabla f(z_t), (\frac{1}{\sqrt{v_{t-1}}+\epsilon} - \frac{1}{\sqrt{v_t}+\epsilon}) \odot m_{t-1}\rangle] \\
&\leq \frac{\eta\beta_1}{1-\beta_1} E[\|\nabla f(z_t)\|\|m_{t-1}\|(\sum_{j=1}^{d} \frac{1}{\sqrt{v_{t-1,j}}+\epsilon} - \frac{1}{\sqrt{v_{t,j}}+\epsilon})] \\
&\leq \frac{\eta\beta_1}{1-\beta_1} GHE[\sum_{j=1}^{d} \frac{1}{\sqrt{v_{t-1,j}}+\epsilon} - \frac{1}{\sqrt{v_{t,j}}+\epsilon}].
\end{aligned}$$

Then we can have

$$E[f(z_{t+1}) - f(z_t)] \leq \frac{\eta\beta_1}{1-\beta_1} GHE[\sum_{j=1}^{d} \frac{1}{\sqrt{v_{t-1,j}}+\epsilon} - \frac{1}{\sqrt{v_{t,j}}+\epsilon}] + \frac{1}{2}L^2\eta^2\mu_2^2(\frac{\beta_1}{1-\beta_1})^2H^2 + \frac{1}{2}\eta^2\mu_2^2H^2 - E[\langle \nabla f(x_t), \frac{\eta}{\sqrt{v_t}+\epsilon} \odot g_t\rangle] + \frac{L\eta^2\beta_1^2H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] + L\eta^2\mu_2^2H^2.$$

By rearranging,

$$E[\langle \nabla f(x_t), \frac{\eta}{\sqrt{v_t}+\epsilon} \odot g_t\rangle] \leq E[f(z_t) - f(z_{t+1})] + \frac{\eta\beta_1}{1-\beta_1} GHE[\sum_{j=1}^{d} \frac{1}{\sqrt{v_{t-1,j}}+\epsilon} - \frac{1}{\sqrt{v_{t,j}}+\epsilon}] + \frac{1}{2}L^2\eta^2\mu_2^2(\frac{\beta_1}{1-\beta_1})^2H^2 + \frac{1}{2}\eta^2\mu_2^2H^2 + \frac{L\eta^2\beta_1^2H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] + L\eta^2\mu_2^2H^2$$

The item on the left hand side also satisfies

$$
\begin{aligned}
E[\langle \nabla f(x_t), \frac{1}{\sqrt{v_t}+\epsilon} \odot g_t \rangle] &\geq E[\sum_{\{j|\nabla f(x_t)_j g_{t,j} \geq 0\}} \mu_1 \nabla f(x_t)_j g_{t,j} + \sum_{\{j|\nabla f(x_t)_j g_{t,j} < 0\}} \mu_2 \nabla f(x_t)_j g_{t,j}] \\
&\geq \sum_{\{j|\nabla f(x_t)_j g_{t,j} \geq 0\}} \mu_1 \nabla f(x_t)_j^2 + \sum_{\{j|\nabla f(x_t)_j g_{t,j} < 0\}} \mu_2 \nabla f(x_t)_j^2 \\
&\geq \mu_1 \|\nabla f(x_t)\|^2
\end{aligned}
$$

Now we obtain:

$$
\begin{aligned}
\eta\mu_1 \|\nabla f(x_t)\|^2 \leq E[f(z_t) - f(z_{t+1})] &+ \frac{\eta\beta_1}{1-\beta_1} GHE[\sum_{j=1}^{d} \frac{1}{\sqrt{v_{t-1,j}}+\epsilon} - \frac{1}{\sqrt{v_{t,j}}+\epsilon}] \\
&+ \frac{1}{2} L^2 \eta^2 \mu_2^2 (\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{1}{2} \eta^2 \mu_2^2 H^2 \\
&+ \frac{L\eta^2 \beta_1^2 H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d} (\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] + L\eta^2 \mu_2^2 H^2
\end{aligned}
$$

Dividing both sides by $\eta\mu_1$ produces:

$$
\begin{aligned}
\|\nabla f(x_t)\|^2 \leq \frac{1}{\eta\mu_1} E[f(z_t) - f(z_{t+1})] &+ \frac{\beta_1}{(1-\beta_1)\mu_1} GHE[\sum_{j=1}^{d} \frac{1}{\sqrt{v_{t-1,j}}+\epsilon} - \frac{1}{\sqrt{v_{t,j}}+\epsilon}] \\
&+ \frac{1}{2\mu_1} L^2 \eta \mu_2^2 (\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{1}{2\mu_1} \eta \mu_2^2 H^2 \\
&+ \frac{L\eta\beta_1^2 H^2}{(1-\beta_1)^2 \mu_1} E[\sum_{j=1}^{d} (\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] + \frac{L\eta\mu_2^2}{\mu_1} H^2
\end{aligned}
$$

Summing from $t = 1$ to $T$, where $T$ is the maximum number of iteration, yields

$$
\begin{aligned}
\sum_{t=1}^{T} [\|\nabla f(x_t)\|^2] \leq \frac{1}{\eta\mu_1} E[f(z_1) - f^*] &+ \frac{\beta_1}{(1-\beta_1)\mu_1} GHE[\sum_{j=1}^{d} \frac{1}{\sqrt{v_{0,j}}+\epsilon} - \frac{1}{\sqrt{v_{T,j}}+\epsilon}] \\
&+ \frac{T}{2\mu_1} L^2 \eta \mu_2^2 (\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{T}{2\mu_1} \eta \mu_2^2 H^2 \\
&+ \frac{L\eta\beta_1^2 H^2}{(1-\beta_1)^2 \mu_1} E[\sum_{j=1}^{d} (\frac{1}{\sqrt{v_{0,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{T,j}}+\epsilon})^2] + \frac{TL\eta\mu_2^2}{\mu_1} H^2
\end{aligned}
$$

Since $v_0 = 0$, $\mu_2 = \frac{1}{\epsilon}$, we have

$$\begin{aligned}\sum_{t=1}^{T}[\|\nabla f(x_t)\|^2] \leq \frac{1}{\eta\mu_1}E[f(z_1) - f^*] + \frac{\beta_1 d}{(1-\beta_1)\mu_1}GH(\mu_2 - \mu_1) \\ + \frac{T}{2\mu_1}L^2\eta\mu_2^2(\frac{\beta_1}{1-\beta_1})^2H^2 + \frac{T}{2\mu_1}\eta\mu_2^2H^2 \\ + \frac{L\eta\beta_1^2 dH^2}{(1-\beta_1)^2\mu_1}(\mu_2^2 - \mu_1^2) + \frac{TL\eta\mu_2^2}{\mu_1}H^2\end{aligned}$$

620

Multiplying the above inequality by $\frac{1}{T}$,

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}[\|\nabla f(x_t)\|^2] &\leq \frac{1}{\eta\mu_1 T}E[f(z_1) - f^*] + \frac{\beta_1 d}{(1-\beta_1)\mu_1 T}GH(\mu_2 - \mu_1) \\ &+ \frac{1}{2\mu_1}L^2\eta\mu_2^2(\frac{\beta_1}{1-\beta_1})^2H^2 + \frac{1}{2\mu_1}\eta\mu_2^2H^2 \\ &+ \frac{L\eta\beta_1^2 dH^2}{(1-\beta_1)^2\mu_1 T}(\mu_2^2 - \mu_1^2) + \frac{L\eta\mu_2^2}{\mu_1}H^2 \\ &\leq \frac{1}{\eta\mu_1 T}E[f(z_1) - f^*] + \frac{\beta_1 d}{(1-\beta_1)\mu_1 T}GH(\mu_2 - \mu_1) \\ &+ (\frac{1}{2\mu_1}L^2\eta\mu_2^2(\frac{\beta_1}{1-\beta_1})^2 + \frac{\eta\mu_2^2}{2\mu_1} + \frac{L\eta\beta_1^2 d(\mu_2^2 - \mu_1^2)}{(1-\beta_1)^2\mu_1 T} + \frac{L\eta\mu_2^2}{\mu_1})H^2\end{aligned}$$

By setting $\eta = \frac{1}{\sqrt{T}}$, let $x_0 = x_1$, then $z_1 = x_1$, $f(z_1) = f(x_1)$ we derive the final result:

$$\begin{aligned}\min_{t=1,\ldots,T}E[\|\nabla f(x_t)\|^2] &\leq \frac{1}{\mu_1\sqrt{T}}E[f(x_1) - f^*] + \frac{\beta_1 d}{(1-\beta_1)\mu_1 T}GH(\mu_2 - \mu_1) \\ &+ (\frac{L^2\mu_2^2}{2\mu_1\sqrt{T}}(\frac{\beta_1}{1-\beta_1})^2 + \frac{\mu_2^2}{2\mu_1\sqrt{T}} + \frac{L\beta_1^2 d(\mu_2^2 - \mu_1^2)}{(1-\beta_1)^2\mu_1 T\sqrt{T}} + \frac{L\mu_2^2}{\mu_1\sqrt{T}})H^2 \\ &= \frac{C_1}{\sqrt{T}} + \frac{C_2}{T} + \frac{C_3}{T\sqrt{T}}\end{aligned}$$

where

$$\begin{aligned}C_1 &= \frac{1}{\mu_1}[f(x_1) - f^*] + (\frac{L^2\mu_2^2}{2\mu_1}(\frac{\beta_1}{1-\beta_1})^2 + \frac{\mu_2^2}{2\mu_1} + \frac{L\mu_2^2}{\mu_1})H^2 \\ C_2 &= \frac{\beta_1(\mu_2 - \mu_1)dGH}{(1-\beta_1)\mu_1}, \\ C_3 &= \frac{L\beta_1^2 dH^2(\mu_2^2 - \mu_1^2)}{(1-\beta_1)^2\mu_1}.\end{aligned}$$

Given $L, G, H, \beta_1$ are constants, we have $C_1 = O(\frac{1}{\epsilon^2})$, $C_2 = O(\frac{d}{\epsilon})$, $C_3 = O(\frac{d}{\epsilon^2})$. Therefore,

$$\min_{t=1,\ldots,T} E[\|\nabla f(x_t)\|^2] \leq O(\frac{1}{\epsilon^2\sqrt{T}} + \frac{d}{\epsilon T} + \frac{d}{\epsilon^2 T\sqrt{T}})$$

□

Thus, we get the sublinear convergence rate of AMSGrad in nonconvex setting, which recovers the well-known result of SGD ([1]) in nonconvex optimization in terms of $T$.

**Remark D.4.9.** *The leading item from the above convergence is $C_1/\sqrt{T}$, $\epsilon$ plays an essential role in the complexity, and we derive a more accurate order $O(\frac{1}{\epsilon^2\sqrt{T}})$. At present, $\epsilon$ is always underestimated and considered to be not associated with accuracy of the solution ([14]). However, it is closely related with complexity, and with bigger $\epsilon$, the computational complexity should be better. This also supports the analysis of A-LR: $\frac{1}{\sqrt{v_t}+\epsilon}$ of* AMSGrad *in our main paper.*

### *D.4.3.* SAMSGrad *Convergence in Nonconvex Setting*

As SAMSGrad also has constrained bound pair $(\mu_3, \mu_4)$, we can learn from the proof of AMSGrad method, which provides us a general framework of such kind of adaptive methods.

Similar to the AMSGrad proof, from L-smoothness and Lemma D.4.6 , we have

*Proof.* From L-smoothness and Lemma D.4.6, we have

$$\begin{aligned} f(z_{t+1}) &\leq f(z_t) + \langle \nabla f(z_t), z_{t+1} - z_t \rangle + \frac{L}{2}\|z_{t+1} - z_t\|^2 \\ &= f(z_t) + \frac{\eta\beta_1}{1-\beta_1}\langle \nabla f(z_t), (\frac{1}{softplus(\sqrt{v_{t-1}})} - \frac{1}{softplus(\sqrt{v_t})}) \odot m_{t-1} \rangle \\ &\quad - \langle \nabla f(z_t), \frac{\eta}{softplus(\sqrt{v_t})} \odot g_t \rangle + \frac{L}{2}\|z_{t+1} - z_t\|^2 \end{aligned}$$

Taking expectation on both sides, and substituting the results of the lemmas, we have

$$
\begin{aligned}
&E[f(z_{t+1}) - f(z_t)] \\
&\leq \frac{\eta\beta_1}{1-\beta_1} E[\langle \nabla f(z_t), (\frac{1}{softplus(\sqrt{v_{t-1}})} - \frac{1}{softplus(\sqrt{v_t})}) \odot m_{t-1} \rangle] \\
&- E[\langle \nabla f(z_t), \frac{\eta}{softplus(\sqrt{v_t})} \odot g_t \rangle] + \frac{L}{2} E[\|z_{t+1} - z_t\|^2] \\
&\leq \frac{\eta\beta_1}{1-\beta_1} E[\langle \nabla f(z_t), (\frac{1}{softplus(\sqrt{v_{t-1}})} - \frac{1}{softplus(\sqrt{v_t})}) \odot m_{t-1} \rangle] \\
&- E[\langle \nabla f(z_t), \frac{\eta}{softplus(\sqrt{v_t})} \odot g_t \rangle] \\
&+ \frac{L\eta^2\beta_1^2 H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d} (\frac{1}{softplus(\sqrt{v_{t-1,j}})})^2 - (\frac{1}{softplus(\sqrt{v_{t,j}})})^2] + L\eta^2\mu_4^2 H^2 \\
&= \frac{\eta\beta_1}{1-\beta_1} GHE[\sum_{j=1}^{d} \frac{1}{softplus(\sqrt{v_{t-1,j}})} - \frac{1}{softplus(\sqrt{v_{t,j}})}] \\
&- E[\langle \nabla f(z_t) - \nabla f(x_t), \frac{\eta}{softplus(\sqrt{v_t})} \odot g_t \rangle] - E[\langle \nabla f(x_t), \frac{\eta}{softplus(\sqrt{v_t})} \odot g_t \rangle] \\
&+ \frac{L\eta^2\beta_1^2 H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d} (\frac{1}{softplus(\sqrt{v_{t-1,j}})})^2 - (\frac{1}{softplus(\sqrt{v_{t,j}})})^2] + L\eta^2\mu_4^2 H^2 \\
&\leq \frac{\eta\beta_1}{1-\beta_1} GHE[\sum_{j=1}^{d} \frac{1}{softplus(\sqrt{v_{t-1,j}})} - \frac{1}{softplus(\sqrt{v_{t,j}})}] \\
&+ \frac{L^2\eta^2\mu_4^2}{2} (\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{\eta^2\mu_4^2}{2} H^2 - E[\langle \nabla f(x_t), \frac{\eta}{softplus(\sqrt{v_t})} \odot g_t \rangle] \\
&+ \frac{L\eta^2\beta_1^2 H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d} (\frac{1}{softplus(\sqrt{v_{t-1,j}})})^2 - (\frac{1}{softplus(\sqrt{v_{t,j}})})^2] + L\eta^2\mu_4^2 H^2
\end{aligned}
$$

By rearranging,

$$
\begin{aligned}
&E[\langle \nabla f(x_t), \frac{\eta}{softplus(\sqrt{v_t})} \odot g_t \rangle] \\
&\leq E[f(z_t) - f(z_{t+1})] + \frac{\eta\beta_1}{1-\beta_1} GHE[\sum_{j=1}^{d} \frac{1}{softplus(\sqrt{v_{t-1,j}})} - \frac{1}{softplus(\sqrt{v_{t,j}})}] \\
&+ \frac{L^2\eta^2\mu_4^2}{2} (\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{\eta^2\mu_4^2}{2} H^2 \\
&+ \frac{L\eta^2\beta_1^2 H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d} (\frac{1}{softplus(\sqrt{v_{t-1,j}})})^2 - (\frac{1}{softplus(\sqrt{v_{t,j}})})^2] + L\eta^2\mu_4^2 H^2
\end{aligned}
$$

The item on the left hand side also satisfies

$$\begin{aligned}E[\langle \nabla f(x_t), \frac{1}{softplus(\sqrt{v_t})} \odot g_t \rangle] &\geq E[\sum_{\{j|\nabla f(x_{t,j})g_{t,j}\geq 0\}} \mu_3 \nabla f(x_{t,j}) g_{t,j} + \sum_{\{j|\nabla f(x_{t,j})g_{t,j}<0\}} \mu_4 \nabla f(x_{t,j}) g_{t,j} \\ &\geq E[\sum_{\{j|\nabla f(x_{t,j})g_{t,j}\geq 0\}} \mu_3 \nabla f(x_{t,j})^2 + \sum_{\{j|\nabla f(x_{t,j})g_{t,j}<0\}} \mu_4 \nabla f(x_{t,j})^2 \\ &\geq \mu_3 \|\nabla f(x_t)\|^2\end{aligned}$$

We then obtain:

$$\begin{aligned}\eta\mu_3\|\nabla f(x_t)\|^2 \leq E[f(z_t) - f(z_{t+1})] &+ \frac{\eta\beta_1}{1-\beta_1} GHE[\sum_{j=1}^{d} \frac{1}{softplus(\sqrt{v_{t-1,j}})} - \frac{1}{softplus(\sqrt{v_{t,j}})}] \\ &+ \frac{L^2\eta^2\mu_4^2}{2}(\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{\eta^2\mu_4^2}{2} H^2 \\ &+ \frac{L\eta^2\beta_1^2 H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d} (\frac{1}{softplus(\sqrt{v_{t-1,j}})})^2 - (\frac{1}{softplus(\sqrt{v_{t,j}})})^2] + L\eta^2\mu_4^2 H^2\end{aligned}$$

Dividing both sides by $\eta\mu_3$ and then summing over $t = 1$ to $T$, brings out

$$\begin{aligned}\sum_{t=1}^{T}[\|\nabla f(x_t)\|^2] \leq \frac{1}{\eta\mu_3} E[f(z_1) - f^*] &+ \frac{\beta_1}{(1-\beta_1)\mu_3} GHE[\sum_{j=1}^{d} \frac{1}{softplus(\sqrt{v_{0,j}})} - \frac{1}{softplus(\sqrt{v_{T,j}})}] \\ &+ \frac{L^2\eta T\mu_4^2}{2\mu_3}(\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{\eta\mu_4^2 T}{2\mu_3} H^2 \\ &+ \frac{L\eta\beta_1^2 H^2}{(1-\beta_1)^2\mu_3} E[\sum_{j=1}^{d} (\frac{1}{softplus(\sqrt{v_{0,j}})})^2 - (\frac{1}{softplus(\sqrt{v_{T,j}})})^2] + \frac{L\eta\mu_4^2 T H^2}{\mu_3}\end{aligned}$$

Because $v_0 = 0$, and $\frac{1}{softplus(0)} = \mu_4$, we have

$$\begin{aligned}\sum_{t=1}^{T}[\|\nabla f(x_t)\|^2] \leq \frac{1}{\eta\mu_3} E[f(z_1) - f^*] &+ \frac{\beta_1 d}{(1-\beta_1)\mu_3} GH(\mu_4 - \mu_3) \\ &+ \frac{L^2\eta T\mu_4^2}{2\mu_3}(\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{\eta\mu_4^2 T}{2\mu_3} H^2 \\ &+ \frac{L\eta\beta_1^2 dH^2}{(1-\beta_1)^2\mu_3}(\mu_4^2 - \mu_3^2) + \frac{L\eta\mu_4^2 T H^2}{\mu_3}\end{aligned}$$

Multiplying both sides by $\frac{1}{T}$ gives,

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}[\|\nabla f(x_t)\|^2] &\leq \frac{1}{\eta\mu_3 T}E[f(z_1)-f^*]+\frac{\beta_1 d}{(1-\beta_1)\mu_3 T}GH(\mu_4-\mu_3)\\ &+\frac{L^2\eta\mu_4^2}{2\mu_3}(\frac{\beta_1}{1-\beta_1})^2H^2+\frac{\eta\mu_4^2}{2\mu_3}H^2\\ &+\frac{L\eta\beta_1^2 dH^2}{(1-\beta_1)^2\mu_3 T}(\mu_4^2-\mu_3^2)+\frac{L\eta\mu_4^2H^2}{\mu_3}\\ &\leq \frac{1}{\eta\mu_3 T}E[f(z_1)-f^*]+\frac{\beta_1 d}{(1-\beta_1)\mu_3 T}GH(\mu_4-\mu_3)\\ &+(\frac{L^2\eta\mu_4^2}{2\mu_3}(\frac{\beta_1}{1-\beta_1})^2+\frac{\eta\mu_4^2}{2\mu_3}+\frac{L\eta\beta_1^2 d}{(1-\beta_1)^2\mu_3 T}(\mu_4^2-\mu_3^2)+\frac{L\eta\mu_4^2}{\mu_3})H^2\end{aligned}$$

Now, we set $\eta=\frac{1}{\sqrt{T}}$, $x_0=x_1$, then $z_1=x_1$, $f(z_1)=f(x_1)$ we derive the final result for the SADAM method:

$$\begin{aligned}\min_{t=1,\ldots,T}E[\|\nabla f(x_t)\|^2] &\leq \frac{1}{\mu_3\sqrt{T}}E[f(x_1)-f^*]+\frac{\beta_1 dGH}{(1-\beta_1)\mu_3 T}(\mu_4-\mu_3)\\ &+(\frac{L^2\mu_4^2}{2\mu_3\sqrt{T}}(\frac{\beta_1}{1-\beta_1})^2+\frac{\mu_4^2}{2\mu_3\sqrt{T}}+\frac{L\beta_1^2 d(\mu_4^2-\mu_3^2)}{(1-\beta_1)^2\mu_3 T\sqrt{T}}+\frac{L\mu_4^2}{\mu_3\sqrt{T}})H^2\\ &=\frac{C_1}{\sqrt{T}}+\frac{C_2}{T}+\frac{C_3}{T\sqrt{T}}\end{aligned}$$

where

$$\begin{aligned}C_1&=\frac{1}{\mu_3}[f(x_1)-f^*]+(\frac{L^2\mu_4^2}{2\mu_3}(\frac{\beta_1}{1-\beta_1})^2+\frac{\mu_4^2}{2\mu_3}+\frac{L\mu_4^2}{\mu_3})H^2\\ C_2&=\frac{\beta_1(\mu_4-\mu_3)dGH}{(1-\beta_1)\mu_3},\\ C_3&=\frac{L\beta_1^2 d(\mu_4^2-\mu_3^2)}{(1-\beta_1)^2\mu_3}H^2.\end{aligned}$$

Given $L, G, H, \beta_1$ are constants, we have $C_1=O(\beta^2)$, $C_2=O(d\beta)$, $C_3=O(d\beta^2)$. Therefore,

$$\min_{t=1,\ldots,T}E[\|\nabla f(x_t)\|^2]\leq O(\frac{\beta^2}{\sqrt{T}}+\frac{d\beta}{T}+\frac{d\beta^2}{T\sqrt{T}})$$

□

Thus, we get the sublinear convergence rate of SAMSGRAD in nonconvex setting, which is the same order of AMSGRAD and recovers the well-known result of SGD [1] in nonconvex optimization in terms of $T$.

**Remark D.4.10.** *The leading item from the above convergence is $C_1/\sqrt{T}$, $\beta$ plays an essential role in the complexity, and a more accurate convergence should be $O(\frac{\beta log(1+e^{\beta})}{\sqrt{T}})$. When $\beta$ is chosen big, this will become $O(\frac{\beta^2}{\sqrt{T}})$, somehow behave like* AMSGRAD*'s case as $O(\frac{1}{\epsilon^2\sqrt{T}})$, which also guides us to have a range of $\beta$; when $\beta$ is chosen small, this will become $O(\frac{1}{\sqrt{T}})$, the computational complexity will get close to SGD case, and $\beta$ is a much smaller number compared with $1/\epsilon$, proving that* SAMSGRAD *converges faster. This also supports the analysis of range of A-LR: $1/softplus(\sqrt{v_t})$ in our main paper.*

*D.4.4. Non-strongly Convex*

In previous works, convex case has been well-studied in adaptive gradient methods. AMSGRAD and later methods PAMSGRAD both use a projection on minimizing objective function, here we want to show a different way of proof in non-strongly convex case. For consistency, we still follow the construction of sequence $\{z_t\}$.

Starting from convexity:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x).$$

Then, for any $x \in \mathbb{R}^d$, $\forall t \in [1, T]$,

$$\langle \nabla f(x), x_t - x^* \rangle \geq f(x_t) - f^*, \tag{15}$$

where $f^* = f(x^*)$, $x^*$ is the optimal solution.

*Proof.* AMSGRAD case:

In the updating rule of AMSGRAD optimizer, $x_{t+1} = x_t - \frac{\eta_t}{\sqrt{v_t}+\epsilon} \odot m_t$, setting stepsize to be fixed, $\eta_t = \eta$, and assume $v_t \geq v_{t-1}$ holds. Using previous results,

$$
\begin{aligned}
&E[\|z_{t+1} - x^*\|^2] \\
&= E[\|z_t + \frac{\eta\beta_1}{1-\beta_1}(\frac{1}{\sqrt{v_{t-1}}+\epsilon} - \frac{1}{\sqrt{v_t}+\epsilon}) \odot m_{t-1} - \frac{\eta}{\sqrt{v_t}+\epsilon} \odot g_t - x^*\|^2] \\
&= E[\|z_t - x^*\|^2] + E[\|\frac{\eta\beta_1}{1-\beta_1}(\frac{1}{\sqrt{v_{t-1}}+\epsilon} - \frac{1}{\sqrt{v_t}+\epsilon}) \odot m_{t-1} - \frac{\eta}{\sqrt{v_t}+\epsilon} \odot g_t\|^2] \\
&+ 2E[\langle \frac{\eta\beta_1}{1-\beta_1}(\frac{1}{\sqrt{v_{t-1}}+\epsilon} - \frac{1}{\sqrt{v_t}+\epsilon}) \odot m_{t-1}, z_t - x^* \rangle] - 2E[\langle \frac{\eta}{\sqrt{v_t}+\epsilon} \odot g_t, z_t - x^* \rangle] \\
&\leq E[\|z_t - x^*\|^2] + 2\frac{\eta^2\beta_1^2}{(1-\beta_1)^2} E[\|(\frac{1}{\sqrt{v_{t-1}}+\epsilon} - \frac{1}{\sqrt{v_t}+\epsilon}) \odot m_{t-1}\|^2] + 2\eta^2 E[\|\frac{1}{\sqrt{v_t}+\epsilon} \odot g_t\|^2] \\
&+ 2\frac{\eta\beta_1}{1-\beta_1} E[\langle (\frac{1}{\sqrt{v_{t-1}}+\epsilon} - \frac{1}{\sqrt{v_t}+\epsilon}) \odot m_{t-1}, z_t - x^* \rangle] - 2\eta E[\langle \frac{1}{\sqrt{v_t}+\epsilon} \odot g_t, z_t - x^* \rangle] \\
&\leq E[\|z_t - x^*\|^2] + 2\frac{\eta^2\beta_1^2 H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d} (\frac{1}{\sqrt{v_{t-1}}+\epsilon})^2 - (\frac{1}{\sqrt{v_t}+\epsilon})^2] + 2\eta^2\mu_2^2 H^2 \\
&+ 2\frac{\eta\beta_1}{1-\beta_1} E[\langle (\frac{1}{\sqrt{v_{t-1}}+\epsilon} - \frac{1}{\sqrt{v_t}+\epsilon}) \odot m_{t-1}, z_t - x^* \rangle] - 2\eta E[\langle \frac{1}{\sqrt{v_t}+\epsilon} \odot g_t, z_t - x^* \rangle]
\end{aligned}
$$

The first inequality holds due to $\|a-b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, the second inequality holds due to D.4.3, D.4.5.

Since $< a, b > \leq \frac{1}{2\eta}a^2 + \frac{\eta}{2}b^2$,

$$
\begin{aligned}
&2E[\langle (\frac{1}{\sqrt{v_{t-1}}+\epsilon} - \frac{1}{\sqrt{v_t}+\epsilon}) \odot m_{t-1}, z_t - x^* \rangle] \\
&\leq \frac{1}{\eta} E[\|(\frac{1}{\sqrt{v_{t-1}}+\epsilon} - \frac{1}{\sqrt{v_t}+\epsilon}) \odot m_{t-1}\|^2] + \eta E[\|z_t - x^*\|^2] \\
&\leq \frac{1}{\eta} H^2 E[\sum_{j=1}^{d} (\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] + \eta E[\|z_t - x^*\|^2]
\end{aligned}
$$

From the definition of $z_t$ and convexity,

$$
\langle \nabla f(x_t), x_t - x^* \rangle \geq f(x_t) - f^* \geq 0
$$

$$
\begin{aligned}
&- 2\eta E[\langle \frac{1}{\sqrt{v_t}+\epsilon} \odot g_t, z_t - x^* \rangle] \\
&= -2\eta E[\langle \frac{1}{\sqrt{v_t}+\epsilon} \odot g_t, x_t - x^* + \frac{\beta_1}{1-\beta_1}(x_t - x_{t-1}) \rangle] \\
&= -2\eta E[\langle \frac{1}{\sqrt{v_t}+\epsilon} \odot g_t, x_t - x^* \rangle] - \frac{2\eta\beta_1}{1-\beta_1} E[\langle \frac{1}{\sqrt{v_t}+\epsilon} \odot g_t, x_t - x_{t-1} \rangle] \\
&= -2\eta E[\langle \frac{1}{\sqrt{v_t}+\epsilon} \odot g_t, x_t - x^* \rangle] - \frac{2\eta^2\beta_1}{1-\beta_1} E[\langle \frac{1}{\sqrt{v_t}+\epsilon} \odot g_t, \frac{1}{\sqrt{v_{t-1}}+\epsilon} \odot m_{t-1} \rangle] \\
&\leq -2\eta\mu_1 \langle \nabla f(x_t), x_t - x^* \rangle + \frac{2\eta^2\beta_1\mu_2^2}{(1-\beta_1)} H^2 \\
&\leq -2\eta\mu_1 (f(x_t) - f^*) + \frac{2\eta^2\beta_1\mu_2^2}{(1-\beta_1)} H^2
\end{aligned}
$$

Plugging in previous two inequalities:

$$
\begin{aligned}
&E[\|z_{t+1} - x^*\|^2] \\
&\leq E[\|z_t - x^*\|^2] + 2\frac{\eta^2\beta_1^2 H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d} (\frac{1}{\sqrt{v_{t-1}}+\epsilon})^2 - (\frac{1}{\sqrt{v_t}+\epsilon})^2] + 2\eta^2\mu_2^2 H^2 \\
&+ \frac{\beta_1 H^2}{1-\beta_1} E[\sum_{j=1}^{d} (\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] + \frac{\eta^2\beta_1}{1-\beta_1} E[\|z_t - x^*\|^2] \\
&- 2\eta\mu_1 (f(x_t) - f^*) + \frac{2\eta^2\beta_1\mu_2^2}{(1-\beta_1)} H^2
\end{aligned}
$$

By rearranging:

$$
\begin{aligned}
&2\eta\mu_1 (f(x_t) - f^*) \\
&\leq E[\|z_t - x^*\|^2] - E[\|z_{t+1} - x^*\|^2] + 2\frac{\eta^2\beta_1^2 H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d} (\frac{1}{\sqrt{v_{t-1}}+\epsilon})^2 - (\frac{1}{\sqrt{v_t}+\epsilon})^2] \\
&+ 2\eta^2\mu_2^2 H^2 + \frac{\beta_1 H^2}{1-\beta_1} E[\sum_{j=1}^{d} (\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] + \frac{\eta^2\beta_1}{1-\beta_1} E[\|z_t - x^*\|^2] \\
&+ \frac{2\eta^2\beta_1\mu_2^2}{(1-\beta_1)} H^2
\end{aligned}
$$

Divide $2\eta\mu_1$ on both sides,

$$f(x_t) - f^* \leq \frac{1}{2\eta\mu_1}(E[\|z_t - x^*\|^2] - E[\|z_{t+1} - x^*\|^2]) + \frac{\eta\beta_1^2 H^2}{(1-\beta_1)^2\mu_1} E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_{t-1}}+\epsilon})^2 - (\frac{1}{\sqrt{v_t}+\epsilon})^2]$$
$$+ \frac{\eta\mu_2^2}{\mu_1}H^2 + \frac{\beta_1 H^2}{2\eta\mu_1(1-\beta_1)} E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2]$$
$$+ \frac{\eta\beta_1}{2\mu_1(1-\beta_1)} E[\|z_t - x^*\|^2] + \frac{\eta\beta_1\mu_2^2}{(1-\beta_1)\mu_1}H^2$$

Assume that $\forall t$, $E[\|x_t - x^*\| \leq D$, for any $m \neq n$, $E[\|x_m - x_n\|] \leq D_\infty$ hold, then $E[\|z_t - x^*\|^2]$ can be bounded.

$$E[\|z_1 - x^*\|^2] = E[\|x_1 - x^*\|^2] \leq D^2 \tag{16}$$
$$E[\|z_t - x^*\|^2] = E[\|x_t - x^* + \frac{\beta_1}{1-\beta_1}(x_t - x_{t-1})\|^2]$$
$$\leq 2E[\|x_t - x^*\|^2] + \frac{2\beta_1^2}{(1-\beta_1)^2}E[\|(x_t - x_{t-1})\|^2]$$
$$\leq 2D^2 + \frac{2\beta_1^2}{(1-\beta_1)^2}D_\infty^2. \tag{17}$$

Thus:

$$f(x_t) - f^* \leq \frac{1}{2\eta\mu_1}(E[\|z_t - x^*\|^2] - E[\|z_{t+1} - x^*\|^2]) + \frac{\eta\beta_1^2 H^2}{(1-\beta_1)^2\mu_1} E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_{t-1}}+\epsilon})^2 - (\frac{1}{\sqrt{v_t}+\epsilon})^2]$$
$$+ \frac{\eta\mu_2^2}{\mu_1}H^2 + \frac{\beta_1 H^2}{2\eta\mu_1(1-\beta_1)} E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2]$$
$$+ \frac{\eta\beta_1 D^2}{\mu_1(1-\beta_1)} + \frac{\eta\beta_1^3 D_\infty^2}{\mu_1(1-\beta_1)^3} + \frac{\eta\beta_1\mu_2^2}{(1-\beta_1)\mu_1}H^2$$

Summing from $t = 1$ to $T$,

$$\sum_{t=1}^{T}(f(x_t)-f^*) \le \frac{1}{2\eta\mu_1}(E[\|z_1-x^*\|^2]-E[\|z_T-x^*\|^2]) + \frac{\eta\beta_1^2H^2}{(1-\beta_1)^2\mu_1}E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_0}+\epsilon})^2-(\frac{1}{\sqrt{v_T}+\epsilon})^2]$$
$$+\frac{\eta\mu_2^2T}{\mu_1}H^2+\frac{\beta_1H^2}{2\eta\mu_1(1-\beta_1)}E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_{0,j}}+\epsilon})^2-(\frac{1}{\sqrt{v_{T,j}}+\epsilon})^2]$$
$$+\frac{\eta\beta_1D^2T}{\mu_1(1-\beta_1)}+\frac{\eta\beta_1^3D_\infty^2T}{\mu_1(1-\beta_1)^3}+\frac{\eta\beta_1\mu_2^2T}{(1-\beta_1)\mu_1}H^2$$
$$\le \frac{1}{2\eta\mu_1}D^2+\frac{\eta\beta_1^2dH^2}{(1-\beta_1)^2\mu_1}(\mu_2^2-\mu_1^2)+\frac{\eta\mu_2^2T}{\mu_1}H^2+\frac{\beta_1dH^2}{2\eta\mu_1(1-\beta_1)}(\mu_2^2-\mu_1^2)$$
$$+\frac{\eta\beta_1D^2T}{\mu_1(1-\beta_1)}+\frac{\eta\beta_1^3D_\infty^2T}{\mu_1(1-\beta_1)^3}+\frac{\eta\beta_1\mu_2^2T}{(1-\beta_1)\mu_1}H^2$$

The second inequality is based on the fact that, when iteration $t$ reaches the maximum number $T$, $x_t$ is the optimal solution, $z_T = x^*$.

By Jensen's inequality,

$$\frac{1}{T}\sum_{t=1}^{T}(f(x_t)-f^*) \ge f(\bar{x}_t)-f^*,$$

where $\bar{x}_t = \frac{1}{T}\sum_{t=1}^{T}x_t$.

Then,

$$f(\bar{x}_t)-f^* \le \frac{D^2}{2\eta\mu_1T}+\frac{\eta\beta_1^2dH^2}{(1-\beta_1)^2\mu_1T}(\mu_2^2-\mu_1^2)+\frac{\eta\mu_2^2}{\mu_1}H^2+\frac{\beta_1dH^2}{2\eta\mu_1(1-\beta_1)T}(\mu_2^2-\mu_1^2)$$
$$+\frac{\eta\beta_1D^2}{\mu_1(1-\beta_1)}+\frac{\eta\beta_1^3D_\infty^2}{\mu_1(1-\beta_1)^3}+\frac{\eta\beta_1\mu_2^2}{(1-\beta_1)\mu_1}H^2$$

By plugging the stepsize $\eta = O(\frac{1}{\sqrt{T}})$, we complete the proof of AMSGrad in non-strongly convex case.

$$f(\bar{x}_t)-f^* \le \frac{D^2}{2\mu_1\sqrt{T}}+\frac{\beta_1^2dH^2}{(1-\beta_1)^2\mu_1T\sqrt{T}}(\mu_2^2-\mu_1^2)+\frac{\mu_2^2}{\mu_1\sqrt{T}}H^2+\frac{\beta_1dH^2}{2\mu_1(1-\beta_1)\sqrt{T}}(\mu_2^2-\mu_1^2)$$
$$+\frac{\beta_1D^2}{\mu_1(1-\beta_1)\sqrt{T}}+\frac{\beta_1^3D_\infty^2}{\mu_1(1-\beta_1)^3\sqrt{T}}+\frac{\beta_1\mu_2^2}{(1-\beta_1)\mu_1\sqrt{T}}H^2$$
$$= O(\frac{1}{\sqrt{T}})+O(\frac{1}{T\sqrt{T}}) = O(\frac{1}{\sqrt{T}}).$$

□

**Remark D.4.11.** *The leading item of convergence order of* AMSGRAD *should be* $O(\frac{\tilde{C}}{\sqrt{T}})$*, where* $\tilde{C} = \frac{D^2}{2\mu_1} + \frac{\mu_2^2}{\mu_1}H^2 + \frac{\beta_1 dH^2}{2\mu_1(1-\beta_1)}(\mu_2^2 - \mu_1^2) + \frac{\beta_1 D^2}{\mu_1(1-\beta_1)} + \frac{\beta_1^3 D_\infty^2}{\mu_1(1-\beta_1)^3} + \frac{\beta_1\mu_2^2}{(1-\beta_1)\mu_1}H^2$*. With fixed* $L, G, H, \beta_1, D, D_\infty$*,* $\tilde{C} = O(\frac{d}{\epsilon^2})$*, which also contains* $\epsilon$ *as well as dimension* $d$*, here with bigger* $\epsilon$*, the order should be better, this also supports the discussion in our main paper.*

The analysis of SAMSGRAD is similar to AMSGRAD, by replacing the bounded pairs $(\mu_1, \mu_2)$ with $(\mu_3, \mu_4)$, we briefly give convergence result below.

*Proof.* SAMSGRAD case:

$$f(\bar{x}_t) - f^* \leq \frac{D^2}{2\eta\mu_3 T} + \frac{\eta\beta_1^2 dH^2}{(1-\beta_1)^2\mu_3 T}(\mu_4^2 - \mu_3^2) + \frac{\eta\mu_4^2}{\mu_3}H^2 + \frac{\beta_1 dH^2}{2\eta\mu_3(1-\beta_1)T}(\mu_4^2 - \mu_3^2) + \frac{\eta\beta_1 D^2}{\mu_3(1-\beta_1)} + \frac{\eta\beta_1^3 D_\infty^2}{\mu_3(1-\beta_1)^3} + \frac{\eta\beta_1\mu_4^2}{(1-\beta_1)\mu_3}H^2$$

By plugging the stepsize $\eta = O(\frac{1}{\sqrt{T}})$, we get the convergence rate of SAMSGRAD in non-strongly convex case.

$$\begin{aligned} f(\bar{x}_t) - f^* &\leq \frac{D^2}{2\mu_3\sqrt{T}} + \frac{\beta_1^2 dH^2}{(1-\beta_1)^2\mu_3 T\sqrt{T}}(\mu_4^2 - \mu_3^2) + \frac{\mu_4^2}{\mu_3\sqrt{T}}H^2 + \frac{\beta_1 dH^2}{2\mu_3(1-\beta_1)\sqrt{T}}(\mu_4^2 - \mu_3^2) \\ &+ \frac{\beta_1 D^2}{\mu_3(1-\beta_1)\sqrt{T}} + \frac{\beta_1^3 D_\infty^2}{\mu_3(1-\beta_1)^3\sqrt{T}} + \frac{\beta_1\mu_4^2}{(1-\beta_1)\mu_3\sqrt{T}}H^2 \\ &= O(\frac{1}{\sqrt{T}}) + O(\frac{1}{T\sqrt{T}}) = O(\frac{1}{\sqrt{T}}). \end{aligned}$$

For brevity,

$$f(\bar{x}_t) - f^* = O(\frac{1}{\sqrt{T}}).$$

□

**Remark D.4.12.** *The leading item of convergence order of* SAMSGRAD *should be* $O(\frac{\tilde{C}}{\sqrt{T}})$*, where* $\tilde{C} = \frac{D^2}{2\mu_3} + \frac{\mu_4^2 d}{\mu_3}H^2 + \frac{\beta_1 dH^2}{2\mu_3(1-\beta_1)}(\mu_4^2 - \mu_3^2) + \frac{\beta_1 D^2}{\mu_3(1-\beta_1)} + \frac{\beta_1^3 D_\infty^2}{\mu_3(1-\beta_1)^3} + \frac{\beta_1\mu_4^2}{(1-\beta_1)\mu_3}H^2$*. With fixed* $L, G, H, \beta_1, D, D_\infty$*,* $\tilde{C} = O(d\beta log(1+e^\beta)) = O(d\beta^2)$*, with small* $\beta$*, the* SAMSGRAD *will be similar to SGD convergence rate, and* $\beta$ *is a much smaller number compared with* $1/\epsilon$*, proving that* SAMSGRAD *method perfoms better than* AMSGRAD *in terms of convergence rate.*

*D.4.5. P-L Condition*

Suppose that strongly convex assumption holds, we can easily deduce the P-L condition (see Lemma D.4.13), which shows that P-L condition is much weaker than strongly convex condition. And we further prove the convergence of ADAM-type optimizer (AMSGRAD and SAMSGRAD) under the P-L condition in non-strongly convex case, which can be extended to the strongly convex case as well.

**Lemma D.4.13.** *Suppose that $f$ is continuously diffentiable and strongly convex with parameter $\gamma$. Then $f$ has the unique minimizer, denoted as $f^* = f(x^*)$. Then for any $x \in \mathbb{R}^d$, we have*

$$\|\nabla f(x)\|^2 \geq 2\gamma(f(x) - f^*).$$

*Proof.* From strongly convex assumption,

$$\begin{aligned} f^* &\geq f(x) + \nabla f(x)^T(x^* - x) + \frac{\gamma}{2}\|x^* - x\|^2 \\ &\geq f(x) + \min_{\xi}(\nabla f(x)^T\xi + \frac{\gamma}{2}\|\xi\|^2) \\ &= f(x) - \frac{1}{2\gamma}\|\nabla f(x)\|^2 \end{aligned}$$

Letting $\xi = x^* - x$, when $\xi = -\frac{\nabla f(x)}{\gamma}$, the quadratic function can achieve its minimum. □

We restate our theorems under PL condition.

**Theorem D.4.14.** *Suppose $f(x)$ satisfies Assumption 1 and PL condition (with parameter $\lambda$) in non-strongly convex case and $v_t \geq v_{t-1}$. Let $\eta_t = \eta = O(\frac{1}{T})$,* AMSGRAD *and* SAMSGRAD *have convergence rate*

$$E[f(x_t) - f^*] \leq O(\frac{1}{T}).$$

*Proof.* AMSGRAD case:

Starting from L-smoothness, and borrowing the previous results we already have

$$\begin{aligned} E[f(z_{t+1}) - f(z_t)] \leq{}& \frac{\eta\beta_1}{1-\beta_1}GHE[\sum_{j=1}^{d}\frac{1}{\sqrt{v_{t-1,j}}+\epsilon} - \frac{1}{\sqrt{v_{t,j}}+\epsilon}] \\ &+ \frac{L^2\eta^2\mu_2^2}{2}(\frac{\beta_1}{1-\beta_1})^2H^2 + \frac{\eta^2\mu_2^2}{2}H^2 - E\langle\nabla f(x_t), \frac{\eta}{\sqrt{v_t}+\epsilon}\odot g_t\rangle \\ &+ \frac{L\eta^2\beta_1^2H^2}{(1-\beta_1)^2}E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] + L\eta^2\mu_2^2H^2 \end{aligned}$$

$$E\langle\nabla f(x_t), \frac{1}{\sqrt{v_t}+\epsilon}\odot g_t\rangle \geq \mu_1\|\nabla f(x_t)\|^2$$

Therefore, we get:

$$E[f(z_{t+1}) - f(z_t)] \leq \frac{\eta\beta_1}{1-\beta_1} GHE[\sum_{j=1}^{d} \frac{1}{\sqrt{v_{t-1,j}} + \epsilon} - \frac{1}{\sqrt{v_{t,j}} + \epsilon}]$$
$$+ \frac{L^2\eta^2\mu_2^2}{2}(\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{\eta^2\mu_2^2}{2} H^2 - \eta\mu_1 \|\nabla f(x_t)\|^2$$
$$+ \frac{L\eta^2\beta_1^2 H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d} (\frac{1}{\sqrt{v_{t-1,j}} + \epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}} + \epsilon})^2] + L\eta^2\mu_2^2 H^2$$

From P-L condition assumption,

$$E[f(z_{t+1})] \leq E[f(z_t)] + \frac{\eta\beta_1}{1-\beta_1} GHE[\sum_{j=1}^{d} \frac{1}{\sqrt{v_{t-1,j}} + \epsilon} - \frac{1}{\sqrt{v_{t,j}} + \epsilon}]$$
$$+ \frac{L^2\eta^2\mu_2^2}{2}(\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{\eta^2\mu_2^2}{2} H^2 - 2\lambda\eta\mu_1 E[f(x_t) - f^*]$$
$$+ \frac{L\eta^2\beta_1^2 H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d} (\frac{1}{\sqrt{v_{t-1,j}} + \epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}} + \epsilon})^2] + L\eta^2\mu_2^2 H^2$$

From convexity,

$$f(z_{t+1}) \geq f(x_{t+1}) + \frac{\beta_1}{1-\beta_1} < \nabla f(x_{t+1}), x_{t+1} - x_t >$$
$$= f(x_{t+1}) + \frac{\beta_1}{1-\beta_1} < \nabla f(x_{t+1}), \frac{\eta}{\sqrt{v_t} + \epsilon} \odot m_t >$$

From L-smoothness,

$$f(z_t) \leq f(x_t) + \frac{\beta_1}{1-\beta_1} < \nabla f(x_t), x_t - x_{t-1} > + \frac{L}{2}(\frac{\beta_1}{1-\beta_1})^2 \|x_t - x_{t-1}\|^2.$$

Then we can obtain

$$
\begin{aligned}
&E[f(x_{t+1})] + \frac{\beta_1}{1-\beta_1} E[< \nabla f(x_{t+1}), \frac{\eta}{\sqrt{v_t}+\epsilon} \odot m_t >] \\
&\leq E[f(x_t)] + \frac{\beta_1}{1-\beta_1} E[< \nabla f(x_t), x_t - x_{t-1} >] + \frac{L}{2}(\frac{\beta_1}{1-\beta_1})^2 E[\|x_t - x_{t-1}\|^2] \\
&+ \frac{\eta\beta_1}{1-\beta_1} GHE[\sum_{j=1}^{d} \frac{1}{\sqrt{v_{t-1,j}}+\epsilon} - \frac{1}{\sqrt{v_{t,j}}+\epsilon}] \\
&+ \frac{L^2\eta^2\mu_2^2}{2}(\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{\eta^2\mu_2^2}{2} H^2 - 2\lambda\eta\mu_1 E[f(x_t) - f^*] \\
&+ \frac{L\eta^2\beta_1^2 H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d} (\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] + L\eta^2\mu_2^2 H^2 \\
&= E[f(x_t)] + \frac{\beta_1}{1-\beta_1} E[< \nabla f(x_t), \frac{\eta}{\sqrt{v_{t-1}}+\epsilon} \odot m_{t-1} >] + \frac{L\eta^2}{2}(\frac{\beta_1}{1-\beta_1})^2 E[\|\frac{1}{\sqrt{v_{t-1}}+\epsilon} \odot m_{t-1}\|^2] \\
&+ \frac{\eta\beta_1}{1-\beta_1} GHE[\sum_{j=1}^{d} \frac{1}{\sqrt{v_{t-1,j}}+\epsilon} - \frac{1}{\sqrt{v_{t,j}}+\epsilon}] \\
&+ \frac{L^2\eta^2\mu_2^2}{2}(\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{\eta^2\mu_2^2}{2} H^2 - 2\lambda\eta\mu_1 E[f(x_t) - f^*] \\
&+ \frac{L\eta^2\beta_1^2 H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d} (\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] + L\eta^2\mu_2^2 H^2
\end{aligned}
$$

By rearranging,

$$
\begin{aligned}
E[f(x_{t+1})] \leq\ & E[f(x_t)] + \frac{\beta_1\eta}{1-\beta_1}(E[< \nabla f(x_t), \frac{1}{\sqrt{v_{t-1}}+\epsilon} \odot m_{t-1} >] - E[< \nabla f(x_{t+1}), \frac{1}{\sqrt{v_t}+\epsilon} \odot m_t >]) \\
&+ \frac{L\eta^2}{2}(\frac{\beta_1}{1-\beta_1})^2 E[\|\frac{1}{\sqrt{v_{t-1}}+\epsilon} \odot m_{t-1}\|^2] + \frac{\eta\beta_1}{1-\beta_1} GHE[\sum_{j=1}^{d} \frac{1}{\sqrt{v_{t-1,j}}+\epsilon} - \frac{1}{\sqrt{v_{t,j}}+\epsilon}] \\
&+ \frac{L^2\eta^2\mu_2^2}{2}(\frac{\beta_1}{1-\beta_1})^2 H^2 + \frac{\eta^2\mu_2^2}{2} H^2 - 2\lambda\eta\mu_1 E[f(x_t) - f^*] \\
&+ \frac{L\eta^2\beta_1^2 H^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d} (\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] + L\eta^2\mu_2^2 H^2
\end{aligned}
$$

From the fact $\pm < a, b > \leq \frac{1}{2}a^2 + \frac{1}{2}b^2$, and Lemma D.4.1, D.4.3,

$$E[<\nabla f(x_t), \frac{1}{\sqrt{v_{t-1}}+\epsilon} \odot m_{t-1}>] = E[<\nabla f(x_{t+1}) \odot \sqrt{\frac{1}{\sqrt{v_{t-1}}+\epsilon}}, m_t \odot \sqrt{\frac{1}{\sqrt{v_{t-1}}+\epsilon}}>]$$
$$\leq \frac{H^2\mu_2}{2} + \frac{H^2\mu_2}{2} \leq H^2\mu_2$$

Similar,

$$-E[<\nabla f(x_{t+1}), \frac{1}{\sqrt{v_t}+\epsilon} \odot m_t>] = -E[<\nabla f(x_{t+1}) \odot \sqrt{\frac{1}{\sqrt{v_{t-1}}+\epsilon}}, m_t \odot \sqrt{\frac{1}{\sqrt{v_{t-1}}+\epsilon}}>]$$
$$\leq \frac{H^2\mu_2}{2} + \frac{H^2\mu_2}{2} \leq H^2\mu_2$$

Then,

$$\begin{aligned}
E[f(x_{t+1})] \leq\ & E[f(x_t)] + \frac{2\beta_1\eta\mu_2}{1-\beta_1}H^2 + \frac{L\eta^2\mu_2^2}{2}(\frac{\beta_1}{1-\beta_1})^2H^2 \\
& + \frac{\eta\beta_1}{1-\beta_1}GHE[\sum_{j=1}^{d}\frac{1}{\sqrt{v_{t-1,j}}+\epsilon} - \frac{1}{\sqrt{v_{t,j}}+\epsilon}] \\
& + \frac{L^2\eta^2\mu_2^2}{2}(\frac{\beta_1}{1-\beta_1})^2H^2 + \frac{\eta^2\mu_2^2}{2}H^2 - 2\lambda\eta\mu_1E[f(x_t) - f^*] \\
& + \frac{L\eta^2\beta_1^2H^2}{(1-\beta_1)^2}E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] + L\eta^2\mu_2^2H^2
\end{aligned}$$

$$
\begin{aligned}
E[f(x_{t+1}) - f^*] &\leq (1 - 2\lambda\eta\mu_1)E[f(x_t) - f^*] + \frac{2\beta_1\eta\mu_2}{1-\beta_1}H^2 + \frac{L\eta^2\mu_2^2}{2}(\frac{\beta_1}{1-\beta_1})^2H^2 \\
&+ \frac{\eta\beta_1}{1-\beta_1}GHE[\sum_{j=1}^{d}\frac{1}{\sqrt{v_{t-1,j}}+\epsilon} - \frac{1}{\sqrt{v_{t,j}}+\epsilon}] \\
&+ \frac{L^2\eta^2\mu_2^2}{2}(\frac{\beta_1}{1-\beta_1})^2H^2 + \frac{\eta^2\mu_2^2}{2}H^2 \\
&+ \frac{L\eta^2\beta_1^2H^2}{(1-\beta_1)^2}E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] + L\eta^2\mu_2^2H^2 \\
&\leq (1 - 2\lambda\eta\mu_1)E[f(x_t) - f^*] + \frac{\eta\beta_1GH}{1-\beta_1}E[\sum_{j=1}^{d}\frac{1}{\sqrt{v_{t-1,j}}+\epsilon} - \frac{1}{\sqrt{v_{t,j}}+\epsilon}] \\
&+ (\frac{2\beta_1\eta\mu_2}{1-\beta_1} + \frac{L\eta^2\mu_2^2}{2}(\frac{\beta_1}{1-\beta_1})^2 + \frac{L^2\eta^2\mu_2^2}{2}(\frac{\beta_1}{1-\beta_1})^2 + \frac{\eta^2\mu_2^2}{2} \\
&+ \frac{L\eta^2\beta_1^2}{(1-\beta_1)^2}E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] + L\eta^2\mu_2^2)H^2
\end{aligned}
$$

Let

$$
\begin{aligned}
\theta &= 1 - 2\lambda\eta\mu_1 \\
\Theta_t &= \frac{\eta\beta_1GH}{1-\beta_1}E[\sum_{j=1}^{d}\frac{1}{\sqrt{v_{t-1,j}}+\epsilon} - \frac{1}{\sqrt{v_{t,j}}+\epsilon}] \\
&+ (\frac{2\beta_1\eta\mu_2}{1-\beta_1} + \frac{L\eta^2\mu_2^2}{2}(\frac{\beta_1}{1-\beta_1})^2 + \frac{L^2\eta^2\mu_2^2}{2}(\frac{\beta_1}{1-\beta_1})^2 + \frac{\eta^2\mu_2^2}{2} \\
&+ \frac{L\eta^2\beta_1^2}{(1-\beta_1)^2}E[\sum_{j=1}^{d}(\frac{1}{\sqrt{v_{t-1,j}}+\epsilon})^2 - (\frac{1}{\sqrt{v_{t,j}}+\epsilon})^2] + L\eta^2\mu_2^2)H^2
\end{aligned}
$$

then we have

$$
E[f(x_{t+1}) - f^*] \leq \theta E[f(x_t) - f^*] + \Theta_t.
$$

Let $\Phi_t = E[f(x_t) - f^*]$, then $\Phi_1 = E[f(x_1) - f^*]$,

$$
\begin{aligned}
\Phi_{t+1} &\leq \theta\Phi_t + \Theta_t \leq \theta^2\Phi_{t-1} + \theta\Theta_{t-1} + \Theta_t \\
&\cdots \\
&\leq \theta^t\Phi_1 + \theta^{t-1}\Theta_1 + \cdots + \theta\Theta_{t-1} + \Theta_t \\
&\overset{\theta<1}{\leq} \theta^t\Phi_1 + \Theta_1 + \cdots + \Theta_{t-1} + \Theta_t.
\end{aligned}
$$

Let $t = T$,

$$
\begin{aligned}
\Phi_{T+1} &\leq \theta^T \Phi_1 + \Theta_1 + \cdots + \Theta_{T-1} + \Theta_T \\
&\leq \theta^T \Phi_1 + \frac{\eta\beta_1 GH}{1-\beta_1} E[\sum_{j=1}^{d} \frac{1}{\sqrt{v_{0,j}} + \epsilon} - \frac{1}{\sqrt{v_{T,j}} + \epsilon}] \\
&+ (\frac{2\beta_1 \eta \mu_2 T}{1-\beta_1} + \frac{L\eta^2 \mu_2^2 T}{2} (\frac{\beta_1}{1-\beta_1})^2 + \frac{L^2 \eta^2 \mu_2^2 T}{2} (\frac{\beta_1}{1-\beta_1})^2 + \frac{\eta^2 \mu_2^2 T}{2} \\
&+ \frac{L\eta^2 \beta_1^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d} (\frac{1}{\sqrt{v_{0,j}} + \epsilon})^2 - (\frac{1}{\sqrt{v_{T,j}} + \epsilon})^2] + L\eta^2 \mu_2^2 T) H^2 \\
&\leq \theta^T \Phi_1 + \frac{\eta\beta_1 GHd}{1-\beta_1} (\mu_2 - \mu_1) + (\frac{2\beta_1 \eta \mu_2 T}{1-\beta_1} + \frac{L\eta^2 \mu_2^2 T}{2} (\frac{\beta_1}{1-\beta_1})^2 + \frac{L^2 \eta^2 \mu_2^2 T}{2} (\frac{\beta_1}{1-\beta_1})^2 \\
&+ \frac{\eta^2 \mu_2^2 T}{2} + \frac{L\eta^2 \beta_1^2 d}{(1-\beta_1)^2} (\mu_2^2 - \mu_1^2) + L\eta^2 \mu_2^2 T) H^2 \\
&= \theta^T \Phi_1 + O(\eta T) + O(\eta^2 T) + O(\eta) + O(\eta^2)
\end{aligned}
$$

From the above inequality, $\eta$ should be set less than $O(\frac{1}{T})$ to ensure all items in the RHS small enough.

Set $\eta = \frac{1}{T^2}$, then $\theta = 1 - 2\lambda\eta\mu_1 = 1 - \frac{2\lambda\mu_1}{T^2}$

$$
\begin{aligned}
\Phi_{T+1} &= \theta^T \Phi_1 + O(\frac{1}{T}) + O(\frac{1}{T^3}) + O(\frac{1}{T^2}) + O(\frac{1}{T^4}) \\
&= \theta^T \Phi_1 + O(\frac{1}{T}) \longrightarrow 0
\end{aligned}
$$

With appropriate $\eta$, we can derive the convergence rate under P-L condition (strongly convex) case.

The proof of SAMSGrad is exactly same as AMSGrad, by replacing the

bounded pairs $(\mu_1, \mu_2)$ with $(\mu_3, \mu_4)$, and we can also get:

$$
\begin{aligned}
\Phi_{T+1} &\leq \theta^T \Phi_1 + \Theta_1 + \cdots + \Theta_{T-1} + \Theta_T \\
&\leq \theta^T \Phi_1 + \frac{\eta\beta_1 GH}{1-\beta_1} E[\sum_{j=1}^{d} \frac{1}{softplus(v_{0,j})} - \frac{1}{softplus(v_{T,j})}] \\
&+ (\frac{2\beta_1\eta\mu_4 T}{1-\beta_1} + \frac{L\eta^2\mu_4^2 T}{2}(\frac{\beta_1}{1-\beta_1})^2 + \frac{L^2\eta^2\mu_4^2 T}{2}(\frac{\beta_1}{1-\beta_1})^2 + \frac{\eta^2\mu_4^2 T}{2} \\
&+ \frac{L\eta^2\beta_1^2}{(1-\beta_1)^2} E[\sum_{j=1}^{d}(\frac{1}{softplus(v_{0,j})})^2 - (\frac{1}{softplus(v_{T,j})})^2] + L\eta^2\mu_4^2 T)H^2 \\
&\leq \theta^T \Phi_1 + \frac{\eta\beta_1 GHd}{1-\beta_1}(\mu_4 - \mu_3) + (\frac{2\beta_1\eta\mu_4 T}{1-\beta_1} + \frac{L\eta^2\mu_4^2 T}{2}(\frac{\beta_1}{1-\beta_1})^2 + \frac{L^2\eta^2\mu_4^2 T}{2}(\frac{\beta_1}{1-\beta_1})^2 \\
&+ \frac{\eta^2\mu_4^2 T}{2} + \frac{L\eta^2\beta_1^2 d}{(1-\beta_1)^2}(\mu_4^2 - \mu_3^2) + L\eta^2\mu_4^2 T)H^2 \\
&= \theta^T \Phi_1 + O(\eta T) + O(\eta^2 T) + O(\eta) + O(\eta^2)
\end{aligned}
$$

By setting appropriate $\eta$, we can also prove the SAMSGRAD converges under PL condition (and strongly convex).

Set $\eta = O(\frac{1}{T^2})$,

$$E[f(x_{T+1}) - f^*] \leq (1 - \frac{2\lambda\mu_3}{T^2})^T E[f(x_1) - f^*] + O(\frac{1}{T}).$$

□

Overall, we have proved AMSGRAD algorithm and SAMSGRAD in all commonly used conditions, our designed algorithms always enjoy the same convergence rate compared with AMSGRAD, and even get better results with appropriate choice of $\beta$ defined in *softplus* function. The proof procedure can be easily extended to other adaptive gradient algorithms, and theoretical results support the discussion and experiments in our main paper.

### *D.4.6. SADAM convergence analysis with VR in Nonconvex Setting*

**Analysis Based on Variance Recursion.** In recent work [19], researchers have proposed a new way to analyze AGMs based on variance recursion, i.e., in terms of $\|m_t - \nabla f(x_t)\|^2$, which has been studied in earlier [32]. Here, we also provide an analysis to prove the convergence to a stationary point with appropriate parameter setting. We list the assumption and key lemmas based on which we prove convergence of our proposed Calibrated AGMs.

An argument $x$ is a $\delta$-first-order stationary point if for an arbitrarily small $\delta > 0$,

$$\|\nabla f(x)\| \leq \delta.$$

**Assumption 2.** *The loss functions $f_i$ and the objective $f$ satisfy:*

*(a)* ***L-smoothness.*** $\forall x, y \in \mathbb{R}^d, \forall i \in \{1, ..., n\}$, $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$.

*(b) The noisy gradient is unbiased and the noise is independent, i.e., $\forall x \in \mathbb{R}^d$, $t \geq 1$, $g_t = \nabla f(x_t) + \xi_t, E[\xi_t] = 0$ and $\xi_i$ is independent of $\xi_j$ if $i \neq j$. And noisy gradient is variance bounded by $E[\|g_t - \nabla f(x_t)\|^2] \leq \sigma^2(1 + c\|\nabla f(x_t)\|^2)$ for some $c \geq 0$.*

In our new analyses framework, our proposed AGMs have bounded A-LR as stated in Lemma 5.1. Thus, we can easily derive $c_l \leq \|\frac{1}{softplus(\sqrt{v_{t,j}})}\|_\infty \leq c_u$ with $0 < c_l < c_u$. We have the following two key lemmas to help us analyze the convergence of SADAM in nonconvex setting.

**Lemma D.4.15.** *[Variance Recursion[32]]*

$$E[\|m_t - \nabla f(x_t)\|^2] \leq \beta_1\|m_{t-1} - \nabla f(x_{t-1})\|^2 + 2(1-\beta_1)^2 E[\|g_t - \nabla f(x_t)\|^2] + \frac{L^2\|x_t - x_{t-1}\|^2}{1-\beta_1}.$$

**Lemma D.4.16.** *[19] If the stepsize $\eta_t \leq \frac{c_l}{2c_u^2 L}$ and $c_l \leq \|\frac{1}{softplus(\sqrt{v_{t,j}})}\|_\infty \leq c_u$ with $0 < c_l < c_u$, we then have*

$$f(x_{t+1}) \leq f(x_t) + \frac{\eta_t c_u}{2}\|\nabla f(x_t) - m_t\|^2 - \frac{\eta_t c_l}{2}\|\nabla f(x_t)\|^2 - \frac{\eta_t c_l}{4}\|m_t\|^2.$$

Based on the above two lemmas, we can derive the following theorem.

**Theorem D.4.17.** *Suppose $f(x)$ is a nonconvex function that satisfies Assumption 2. Let $\Delta_t = \|m_t - \nabla f(x_t)\|^2$, with $1-\beta_1 \leq \frac{\delta^2 c_l}{12\sigma^2 c_u}$, $T \geq \max\{\frac{6\Delta_1 c_u}{(1-\beta_1)\delta^2 c_l}, \frac{12(f(x_1)-f^*)}{\eta_t \delta^2 c_l}\}$, then SADAM method has $E[\frac{1}{T}\sum_{t=1}^T \|\nabla f(x_t)\|^2] \leq \delta^2$.*

*Proof.* From Lemma D.4.15, we have

$$\begin{aligned}
&E[\|m_t - \nabla f(x_t)\|^2] \\
&\leq \beta_1\|m_{t-1} - \nabla f(x_{t-1})\|^2 + 2(1-\beta_1)^2 E[\|g_t - \nabla f(x_t)\|^2] + \frac{L^2\|x_t - x_{t-1}\|^2}{1-\beta_1} \\
&\leq \beta_1\|m_{t-1} - \nabla f(x_{t-1})\|^2 + 2(1-\beta_1)^2\sigma^2(1 + c\|\nabla f(x_t)\|^2) + \frac{L^2\|x_t - x_{t-1}\|^2}{1-\beta_1}
\end{aligned}$$

Then, we can get

$$\begin{aligned}
E[\|m_{t-1} - \nabla f(x_{t-1})\|^2] \leq E[&\frac{\|m_{t-1} - \nabla f(x_{t-1})\|^2 - \|m_t - \nabla f(x_t)\|^2}{1-\beta_1}] \\
&+ 2(1-\beta_1)^2\sigma^2(1 + c\|\nabla f(x_t)\|^2) + \frac{L^2\|x_t - x_{t-1}\|^2}{1-\beta_1}
\end{aligned}$$

Let $\Delta_t = \|m_t - \nabla f(x_t)\|^2$. Summing $\Delta$ over $t$ from 1 to $T$ yields,

$$E[\sum_{t=1}^{T}\Delta_{t-1}] \le E[\sum_{t=1}^{T}\frac{\Delta_{t-1}-\Delta_t}{1-\beta_1} + 2(1-\beta_1)^2\sigma^2 T + 2(1-\beta_1)^2\sigma^2 c\sum_{t=1}^{T}\|\nabla f(x_t)\|^2 + \sum_{t=1}^{T}\frac{L^2\eta_t^2 c_u^2\|m_{t-1}\|^2}{1-\beta_1}]$$

From Lemma D.4.16, we have

$$\begin{aligned}
\frac{\eta_t c_l}{2}\sum_{t=1}^{T}E[\|\nabla f(x_t)\|^2] &\le \sum_{t=1}^{T}[f(x_t)-f(x_{t+1})] + \frac{\eta_t c_u}{2}\|\nabla f(x_t)-m_t\|^2 - \frac{\eta_t c_l}{4}\sum_{t=1}^{T}\|m_t\|^2 \\
&\le \sum_{t=1}^{T}[f(x_t)-f(x_{t+1})] + \frac{\eta_t c_u}{2}(E[\sum_{t=1}^{T}\frac{\Delta_t-\Delta_{t+1}}{1-\beta_1} + 2(1-\beta_1)^2\sigma^2 T \\
&\quad + 2(1-\beta_1)^2\sigma^2 c\sum_{t=1}^{T}\|\nabla f(x_{t+1})\|^2 + \sum_{t=1}^{T}\frac{L^2\eta_t^2 c_u^2\|m_t\|^2}{1-\beta_1}]) - \frac{\eta_t c_l}{4}\sum_{t=1}^{T}\|m_t\|^2 \\
&\le f(x_1) - f^* + \frac{\eta_t c_u}{2}(E[\sum_{t=1}^{T}\frac{\Delta_t-\Delta_{t+1}}{1-\beta_1} + 2(1-\beta_1)^2\sigma^2 T \\
&\quad + 4(1-\beta_1)^2\sigma^2 c\sum_{t=1}^{T}\|\nabla f(x_t)\|^2 + 4(1-\beta_1)^2\sigma^2 c\frac{L^2\eta_t^2 c_u^2\|m_t\|^2}{1-\beta_1} \\
&\quad + \sum_{t=1}^{T}\frac{L^2\eta_t^2 c_u^2\|m_t\|^2}{1-\beta_1}]) - \frac{\eta_t c_l}{4}\sum_{t=1}^{T}\|m_t\|^2
\end{aligned}$$

Let $L^2\eta_t^2 c_u^3/(2(1-\beta_1)^2) \le c_l/8, i.e., \eta_t \le \frac{(1-\beta_1)\sqrt{c_l}}{2L\sqrt{c_u^3}}$ and $2(1-\beta)\sigma^2 c \le c_l/(4c_u)$, $2(1-\beta)\sigma^2 c L^2\eta_t^2 c_u^3 \le c_l/8, i.e., \eta_t \le \frac{1}{\sqrt{2}c_u L}$,

$$\frac{1}{T}\sum_{t=1}^{T}E[\|\nabla f(x_t)\|^2] \le \frac{2}{\eta_t c_l T}[f(x_1)-f^*] + \frac{c_u}{(1-\beta_1)Tc_l}\Delta_1 + 2(1-\beta_1)^2\sigma^2\frac{c_u}{c_l} + \frac{1}{2T}\sum_{t=1}^{T}E[\|\nabla f(x_t)\|^2]$$

Then we get

$$\frac{1}{T}\sum_{t=1}^{T}E[\|\nabla f(x_t)\|^2] \le \frac{4}{\eta_t c_l T}[f(x_1)-f^*] + 2\frac{c_u}{(1-\beta_1)Tc_l}\Delta_1 + 4(1-\beta_1)^2\sigma^2\frac{c_u}{c_l}$$

With appropriate choice of $\beta_1$ and T, we can show the calibrated AGMs converge to $\delta-$first-order stationary point, i.e., if $1-\beta_1 \le \frac{\delta^2 c_l}{12\sigma^2 c_u}$, $T \ge \max\{\frac{6\Delta_1 c_u}{(1-\beta_1)\delta^2 c_l}, \frac{12(f(x_1)-f^*)}{\eta_t\delta^2 c_l}\}$ are satisfied, then $E[\frac{1}{T}\sum_{t=1}^{T}\|\nabla f(x_t)\|^2] \le \delta^2$ holds. □