
An Optimistic Acceleration of AMSGrad for Nonconvex Optimization

Anonymous Author(s)

Affiliation

Address

email

Abstract

We propose a new variant of AMSGrad [Reddi et al., 2018], a popular adaptive gradient based optimization algorithm widely used for training deep neural networks. Our algorithm adds prior knowledge about the sequence of consecutive mini-batch gradients and leverages its underlying structure making the gradients sequentially predictable. By exploiting the predictability and ideas from optimistic online learning, the proposed algorithm can accelerate the convergence and increase sample efficiency. After establishing a tighter upper bound under some convexity conditions on the regret, we offer a complimentary view of our algorithm which generalizes the offline and stochastic version of nonconvex optimization. In the nonconvex case, we establish a non-asymptotic convergence bound independently of the initialization. We illustrate the practical speedup on several deep learning models via numerical experiments.

1 Introduction

Deep learning models have been successful in several applications, from robotics (e.g., [Levine et al., 2017]), computer vision (e.g. [He et al., 2016, Goodfellow et al., 2014]), reinforcement learning (e.g., [Mnih et al., 2013]) and natural language processing (e.g., [Graves et al., 2013]). With the sheer size of modern datasets and the dimension of neural networks, speeding up training is of utmost importance. To do so, several algorithms have been proposed in recent years, such as AMSGRAD [Reddi et al., 2018], ADAM [Kingma and Ba, 2015], RMSPROP [Tieleman and Hinton, 2012], ADADELTA [Zeiler, 2012], and NADAM [Dozat, 2016]. All the prevalent algorithms for training deep networks mentioned above combine two ideas: the idea of adaptivity from ADAGRAD [Duchi et al., 2011, McMahan and Streeter, 2010] and the idea of momentum from NESTEROV’S METHOD [Nesterov, 2004] or HEAVY BALL method [Polyak, 1964]. ADAGRAD is an online learning algorithm that works well compared to the standard online gradient descent when the gradient is sparse. Its update has a notable feature: it leverages an anisotropic learning rate depending on the magnitude of the gradient for each dimension which helps in exploiting the geometry of the data. On the other hand, NESTEROV’S METHOD or HEAVY BALL Method [Polyak, 1964] is an accelerated optimization algorithm which update not only depends on the current iterate and gradient but also depends on the past gradients (i.e. momentum). State-of-the-art algorithms such as AMSGRAD [Reddi et al., 2018] and ADAM [Kingma and Ba, 2015] leverage these ideas to accelerate the training of nonconvex objective functions, for instance deep neural networks losses.

In this paper, we propose an algorithm that goes beyond the hybrid of the adaptivity and momentum approach. Our algorithm is inspired by OPTIMISTIC ONLINE LEARNING [Chiang et al., 2012, Rakhlin and Sridharan, 2013, Syrgkanis et al., 2015, Abernethy et al., 2018, Mertikopoulos et al., 2018], which assumes that, in each round of online learning, a *predictable process* of the gradient of the loss function is available. Then, an action is played exploiting these predictors. By capitalizing on this (possibly) arbitrary process, algorithms in OPTIMISTIC ONLINE LEARNING enjoy smaller

38 regret than the ones without gradient predictions. We combine the OPTIMISTIC ONLINE LEARNING
39 idea with the adaptivity and the momentum ideas to design a new algorithm — OPT-AMSGRAD.

40 A single work along that direction stands out. [Daskalakis et al. \[2018\]](#) develop OPTIMISTIC-ADAM
41 leveraging optimistic online mirror descent [[Rakhlin and Sridharan, 2013](#)]. Yet, OPTIMISTIC-
42 ADAM is specifically designed to optimize two-player games, e.g., GANs [[Goodfellow et al., 2014](#)]
43 which is in particular a two-player zero-sum game. There have been some related works in OP-
44 TIMISTIC ONLINE LEARNING [[Chiang et al., 2012](#), [Rakhlin and Sridharan, 2013](#), [Syrkanis et al.,](#)
45 [2015](#)] showing that if both players use an OPTIMISTIC type of update, then accelerating the con-
46 vergence to the equilibrium of the game is possible. [Daskalakis et al. \[2018\]](#) build on these related
47 works and show that OPTIMISTIC-MIRROR-DESCENT can avoid the cycle behavior in a bilinear
48 zero-sum game accelerating the convergence. In contrast, in this paper, the proposed algorithm is
49 designed to accelerate nonconvex optimization (e.g., empirical risk minimization). To the best of our
50 knowledge, this is the first work exploring towards this direction and bridging the unfilled *theoretical*
51 gap at the crossroads of online learning and stochastic optimization.

52 The **contributions** of our paper are as follows:

- 53 • We derive an optimistic variant of AMSGRAD borrowing techniques from online learning
54 procedures. Our method relies on (I) the addition of *prior knowledge* in the sequence
55 of model parameter estimates leveraging a predictable process able to provide guesses of
56 gradients through the iterations; (II) the construction of a *double update* algorithm done
57 sequentially. We interpret this two-projection step as the learning of the global parameter
58 and of an underlying scheme which makes the gradients sequentially predictable.
- 59 • We focus on the *theoretical* justifications of our method by establishing novel *non-*
60 *asymptotic* and *global* convergence rates in both convex and nonconvex cases. Based on
61 *convex regret minimization* and *nonconvex stochastic optimization* views, we prove, re-
62 spectively, that our algorithm suffers regret of $\mathcal{O}(\sqrt{\sum_{t=1}^T \|g_t - m_t\|_{\psi_{t-1}}^2})$ and achieves a
63 convergence rate $\mathcal{O}(\sqrt{d/T} + d/T)$, where g_t is the gradient and m_t is its prediction.

64 The proposed algorithm adapts to the informative dimensions, exhibits momentum, and also ex-
65 ploits a good guess of the next gradient to facilitate acceleration. Besides the complete convergence
66 analysis of OPT-AMSGRAD, we conduct numerical experiments and show that the proposed algo-
67 rithm not only accelerates the training procedure, but also leads to better empirical generalization
68 performance.

69 **Notations:** We follow the notations of adaptive optimization [[Kingma and Ba, 2015](#), [Reddi et al.,](#)
70 [2018](#)]. For any $u, v \in \mathbb{R}^d$, u/v represents the element-wise division, u^2 the element-wise square, \sqrt{u}
71 the element-wise square-root. We denote $g_{1:T}[i]$ as the sum of the i_{th} element of $g_1, \dots, g_T \in \mathbb{R}^d$
72 and $\|\cdot\|$ as the Euclidean norm.

73 2 Preliminaries

Optimistic Online learning. The standard setup of ONLINE LEARNING is that, in each round t , an
online learner selects an action $w_t \in \Theta \subseteq \mathbb{R}^d$, observes $\ell_t(\cdot)$ and suffers the associated loss $\ell_t(w_t)$
after the action is committed. The goal is to minimize the regret,

$$\mathcal{R}_T(\{w_t\}) := \sum_{t=1}^T \ell_t(w_t) - \sum_{t=1}^T \ell_t(w^*),$$

74 which is the cumulative loss of the learner minus the cumulative loss of some benchmark $w^* \in \Theta$.
75 The idea of OPTIMISTIC ONLINE LEARNING (e.g., [[Chiang et al., 2012](#), [Rakhlin and Sridharan,](#)
76 [2013](#), [Syrkanis et al., 2015](#), [Abernethy et al., 2018](#)]) is as follows. In each round t , the learner
77 exploits a guess $m_t(\cdot)$ of the gradient $\nabla \ell_t(\cdot)$ to choose an action w_t ¹. Consider the FOLLOW-THE-
78 REGULARIZED-LEADER (FTRL, [[Hazan, 2016](#)]) online learning algorithm which update reads

$$w_t = \arg \min_{w \in \Theta} \langle w, L_{t-1} \rangle + \frac{1}{\eta} \mathbf{R}(w),$$

¹Imagine that if the learner would have known $\nabla \ell_t(\cdot)$ (i.e., exact guess) before committing its action, then
it would exploit the knowledge to determine its action and consequently minimize the regret.

where η is a parameter, $R(\cdot)$ is a 1-strongly convex function with respect to a given norm on the constraint set Θ , and $L_{t-1} := \sum_{s=1}^{t-1} g_s$ is the cumulative sum of gradient vectors of the loss functions up to round $t - 1$. It has been shown that FTRL has regret at most $\mathcal{O}(\sqrt{\sum_{t=1}^T \|g_t\|_*^2})$. The update of its optimistic variant, called OPTIMISTIC-FTRL and developed in [Syrkanis et al., 2015] reads

$$w_t = \arg \min_{w \in \Theta} \langle w, L_{t-1} + m_t \rangle + \frac{1}{\eta} R(w), \quad (1)$$

where $\{m_t\}_{t>0}$ is a predictable process incorporating (possibly arbitrary) knowledge about the sequence of gradients $\{g_t := \nabla \ell_t(w_t)\}_{t>0}$. Under the assumption that the loss functions are convex, it has been shown in [Syrkanis et al., 2015] that the regret of OPTIMISTIC-FTRL is at most $\mathcal{O}(\sqrt{\sum_{t=1}^T \|g_t - m_t\|_*^2})$.

Remark: Note that the usual worst-case bound is preserved even when the predictors $\{m_t\}_{t>0}$ do not predict well the gradients. Indeed, if we take the example of OPTIMISTIC-FTRL, the bound reads $\sqrt{\sum_{t=1}^T \|g_t - m_t\|_*^2} \leq 2 \max_{w \in \Theta} \|\nabla \ell_t(w)\| \sqrt{T}$ which is equal to the usual bound up to a factor 2 [Rakhlin and Sridharan, 2013], under certain boundedness assumptions on Θ detailed below. Yet, when the predictors $\{m_t\}_{t>0}$ are well designed, the resulting regret will be lower. We will have a similar argument when comparing OPT-AMSGRAD and AMSGRAD regret bounds in Section 4.1.

We emphasize, in Section 3, the importance of leveraging a good guess m_t for updating w_t in order to get a fast convergence rate (or equivalently, small regret) and introduce in Section 6 a simple predictable process $\{m_t\}_{t>0}$ leading to empirical acceleration on various applications.

Adaptive optimization methods. Adaptive optimization has been popular in various deep learning applications due to their superior empirical performance. ADAM [Kingma and Ba, 2015], a popular adaptive algorithm, combines momentum [Polyak, 1964] and anisotropic learning rate of ADAGRAD [Duchi et al., 2011]. More specifically, the learning rate of ADAGRAD at time T for dimension j is proportional to the inverse of $\sqrt{\sum_{t=1}^T g_t[j]^2}$, where $g_t[j]$ is the j -th element of the gradient vector g_t at time t .

Algorithm 1 AMSGRAD [Reddi et al., 2018]

```

1: Required: parameter  $\beta_1, \beta_2$ , and  $\eta_t$ .
2: Init:  $w_1 \in \Theta \subseteq \mathbb{R}^d$  and  $v_0 = \epsilon \mathbf{1} \in \mathbb{R}^d$ .
3: for  $t = 1$  to  $T$  do
4:   Get mini-batch stochastic gradient  $g_t$  at  $w_t$ .
5:    $\theta_t = \beta_1 \theta_{t-1} + (1 - \beta_1) g_t$ .
6:    $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$ .
7:    $\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$ .
8:    $w_{t+1} = w_t - \eta_t \frac{\theta_t}{\sqrt{\hat{v}_t}}$ . (element-wise division)
9: end for

```

This adaptive learning rate helps accelerating the convergence when the gradient vector is sparse [Duchi et al., 2011], yet, when applying ADAGRAD to train deep neural networks, it is observed that the learning rate might decay too fast, see [Kingma and Ba, 2015] for more details. Therefore, Kingma and Ba [2015] put forward ADAM that uses a moving average of the gradients divided by the square root of the second moment of this moving average (element-wise multiplication), for updating the model parameter w . A variant, called AMSGRAD and detailed in Algorithm 1, has been developed in [Reddi et al., 2018] to fix ADAM failures. The difference between ADAM and AMSGRAD lies in Line 7 of Algorithm 1. The AMSGRAD algorithm [Reddi et al., 2018] applies the \max operation on the second moment to guarantee a non-increasing learning rate $\eta_t / \sqrt{\hat{v}_t}$, which helps for the convergence (i.e. average regret $\mathcal{R}_T / T \rightarrow 0$).

3 OPT-AMSGRAD Algorithm

We formulate in this section the proposed optimistic acceleration of AMSGrad, namely OPT-AMSGRAD, and detailed in Algorithm 2. It combines the idea of *adaptive optimization* with *optimistic learning*. At each iteration, the learner computes a gradient vector $g_t := \nabla \ell_t(w_t)$ at w_t (line 4), then maintains an exponential moving average of $\theta_t \in \mathbb{R}^d$ (line 5) and $v_t \in \mathbb{R}^d$ (line

Algorithm 2 OPT-AMSGRAD

```

1: Required: parameter  $\beta_1, \beta_2, \epsilon$ , and  $\eta_t$ .
2: Init:  $w_1 = w_{-1/2} \in \Theta \subseteq \mathbb{R}^d$  and  $v_0 = \epsilon 1 \in \mathbb{R}^d$ .
3: for  $t = 1$  to  $T$  do
4:   Get mini-batch stochastic gradient  $g_t$  at  $w_t$ .
5:    $\theta_t = \beta_1 \theta_{t-1} + (1 - \beta_1) g_t$ .
6:    $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$ .
7:    $\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$ .
8:    $\tilde{w}_{t+1} = \tilde{w}_t - \eta_t \frac{\theta_t}{\sqrt{\hat{v}_t}}$ .
9:    $w_{t+1} = \tilde{w}_{t+1} - \eta_t \frac{h_{t+1}}{\sqrt{\hat{v}_t}}$ ,
      where  $h_{t+1} := \beta_1 \theta_{t-1} + (1 - \beta_1) m_{t+1}$  with  $m_{t+1}$  the guess of  $g_{t+1}$ .
10: end for

```

6), which is followed by the \max operation to get $\hat{v}_t \in \mathbb{R}^d$ (line 7). The learner first updates an auxiliary variable $\tilde{w}_{t+1} \in \Theta$ (line 8), then computes the next model parameter w_{t+1} (line 9). Observe that the proposed algorithm does not reduce to AMSGRAD when $m_t = 0$, contrary to the optimistic variant of FTRL. Furthermore, combining line 8 and line 9 yields the following single step $w_{t+1} = \tilde{w}_t - \eta_t(\theta_t + h_{t+1})/\sqrt{\hat{v}_t}$.

Compared to AMSGRAD, the algorithm is characterized by a *two-level* update that interlinks some auxiliary state \tilde{w}_t and the model parameter state, w_t , similarly to the OPTIMISTIC MIRROR DESCENT algorithm developed in [Rakhlin and Sridharan, 2013]. It leverages the auxiliary variable (hidden model) to update and commit w_{t+1} , which exploits the guess m_{t+1} , see Figure 1.

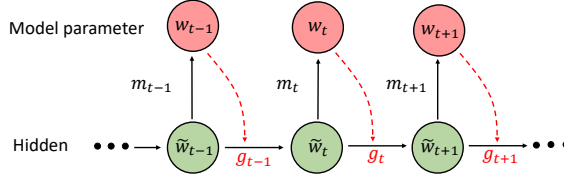


Figure 1: OPT-AMSGRAD underlying structure.

In the following analysis, we show that this interleaving actually leads to some cancellation in the regret bound. Such two-levels method where the guess m_t is equal to the last known gradient g_{t-1} has been exhibited recently in [Chiang et al., 2012]. The gradient prediction process plays an important role as discussed in Section 6. The proposed OPT-AMSGRAD algorithm inherits three properties: (i) Adaptive learning rate of each dimension as ADAGRAD [Duchi et al., 2011] (line 6, line 8 and line 9). (ii) Exponential moving average of the past gradients as NESTEROV'S METHOD [Nesterov, 2004] and the HEAVY-BALL method [Polyak, 1964] (line 5). (iii) Optimistic update that exploits *prior knowledge* of the next gradient vector as in optimistic online learning algorithms [Chiang et al., 2012, Rakhlin and Sridharan, 2013, Syrgkanis et al., 2015] (line 9). The first property helps for acceleration when the gradient has a sparse structure. The second one is from the long-established idea of momentum which can also help for acceleration. The last property can lead to an acceleration when the prediction of the next gradient is good as mentioned above when introducing the regret bound for the OPTIMISTIC-FTRL algorithm. This property will be elaborated whilst establishing the theoretical analysis of OPT-AMSGRAD.

4 Convergence Analysis

More notations. We denote the Mahalanobis norm by $\|\cdot\|_H := \sqrt{\langle \cdot, H \cdot \rangle}$ for some positive semidefinite (PSD) matrix H . We let $\psi_t(x) := \langle x, \text{diag}\{\hat{v}_t\}^{1/2} x \rangle$ for a PSD matrix $H_t^{1/2} := \text{diag}\{\hat{v}_t\}^{1/2}$, where $\text{diag}\{\hat{v}_t\}$ represents the diagonal matrix which i_{th} diagonal element is $\hat{v}_t[i]$ defined in Algorithm 2. We define its corresponding Mahalanobis norm by $\|\cdot\|_{\psi_t} := \sqrt{\langle \cdot, \text{diag}\{\hat{v}_t\}^{1/2} \cdot \rangle}$, where we abuse the notation ψ_t to represent the PSD matrix $H_t^{1/2} := \text{diag}\{\hat{v}_t\}^{1/2}$. Note that $\psi_t(\cdot)$ is 1-strongly convex with respect to the norm $\|\cdot\|_{\psi_t}$, i.e., $\psi_t(\cdot)$

147 satisfies $\psi_t(u) \geq \psi_t(v) + \langle \psi_t(v), u - v \rangle + \frac{1}{2} \|u - v\|_{\psi_t}^2$ for any point $(u, v) \in \Theta^2$. A consequence
 148 of 1-strong convexity of $\psi_t(\cdot)$ is that $B_{\psi_t}(u, v) \geq \frac{1}{2} \|u - v\|_{\psi_t}^2$, where the Bregman divergence
 149 $B_{\psi_t}(u, v)$ is defined as $B_{\psi_t}(u, v) := \psi_t(u) - \psi_t(v) - \langle \psi_t(v), u - v \rangle$ with $\psi_t(\cdot)$ as the distance
 150 generating function. We also define the corresponding dual norm $\|\cdot\|_{\psi_t^*} := \sqrt{\langle \cdot, \text{diag}\{\hat{v}_t\}^{-1/2} \cdot \rangle}$.
 151 The proofs of the results are deferred to the Appendix.

152 4.1 Convex Regret Analysis

153 In the following, we assume convexity of $\{\ell_t\}_{t>0}$ and that Θ has a bounded diameter D_∞ , which
 154 is a standard assumption for adaptive methods [Reddi et al., 2018, Kingma and Ba, 2015] and is
 155 necessary in regret analysis.

156 **Theorem 1.** *Suppose the learner incurs a sequence of convex loss functions $\{\ell_t(\cdot)\}$. Then, OPT-
 157 AMSGRAD (Algorithm 2) has regret*

$$\begin{aligned} \mathcal{R}_T \leq & \frac{B_{\psi_1}(w^*, \tilde{w}_1)}{\eta_1} + \sum_{t=1}^T \frac{\eta_t}{2} \|g_t - \tilde{m}_t\|_{\psi_{t-1}^*}^2 \\ & + \frac{D_\infty^2}{\eta_{\min}} \sum_{i=1}^d \hat{v}_T^{1/2}[i] + D_\infty^2 \beta_1^2 \sum_{t=1}^T \|g_t - \theta_{t-1}\|_{\psi_{t-1}^*}^2, \end{aligned}$$

158 where $\tilde{m}_{t+1} = \beta_1 \theta_{t-1} + (1 - \beta_1) m_{t+1}$, $g_t := \nabla \ell_t(w_t)$, $\eta_{\min} := \min_t \eta_t$ and D_∞^2 is the diameter of
 159 the bounded set Θ . The result holds for any benchmark $w^* \in \Theta$ and any step size sequence $\{\eta_t\}_{t>0}$.

160 **Corollary 1.** *Suppose $\beta_1 = 0$ and $\{v_t\}_{t>0}$ is a monotonically increasing sequence, then we obtain
 161 the following regret bound for any $w^* \in \Theta$ and sequence of stepsizes $\{\eta_t = \eta/\sqrt{t}\}_{t>0}$:*

$$\begin{aligned} \mathcal{R}_T \leq & \frac{B_{\psi_1}}{\eta_1} + \frac{\eta \sqrt{1 + \log T}}{\sqrt{1 - \beta_2}} \sum_{i=1}^d \|(g - m)_{1:T}[i]\|_2 \\ & + \frac{D_\infty^2}{\eta_{\min}} \sum_{i=1}^d \left[(1 - \beta_2) \sum_{s=1}^T \beta_2^{T-s} g_s^2[i] \right]^{1/2}, \end{aligned}$$

162 where $B_{\psi_1} := B_{\psi_1}(w^*, \tilde{w}_1)$, $g_t := \nabla \ell_t(w_t)$ and $\eta_{\min} := \min_t \eta_t$.

163 We can compare the bound of Corollary 1 with that of AMSGRAD [Reddi et al., 2018] with $\eta_t =$
 164 η/\sqrt{t} :

$$\mathcal{R}_T \leq \frac{\eta \sqrt{1 + \log T}}{\sqrt{1 - \beta_2}} \sum_{i=1}^d \|g_{1:T}[i]\|_2 + \frac{\sqrt{T}}{2\eta} D_\infty^2 \sum_{i=1}^d \hat{v}_T[i]^2. \quad (2)$$

165 For convex regret minimization, Corollary 1 yields a regret of $\mathcal{O}(\sqrt{\sum_{t=1}^T \|g_t - m_t\|_{\psi_{t-1}^*}^2})$ with an
 166 access to an arbitrary predictable process $\{m_t\}_{t>0}$ of the mini-batch gradients. We notice from
 167 the second term in Corollary 1 compared to the first term in (2) that better predictors lead to lower
 168 regret. The construction of the predictions $\{m_t\}_{t>0}$ is thus of utmost importance for achieving
 169 optimal acceleration and can be learned through the iterations [Rakhlin and Sridharan, 2013]. In
 170 Section 6, we derive a basic, yet effective, gradient prediction algorithm, see Algorithm 4, embedded
 171 in OPT-AMSGRAD.

172 4.2 Finite-Time Analysis in the Nonconvex Case

173 We discuss the offline and stochastic nonconvex optimization properties of our online framework.
 174 As stated in the introduction, this paper is about solving optimization problems instead of solving
 175 zero-sum games. Classically, the optimization problem we are tackling reads:

$$\min_{w \in \Theta} f(w) := \mathbb{E}[f(w, \xi)] = n^{-1} \sum_{i=1}^n \mathbb{E}[f(w, \xi_i)], \quad (3)$$

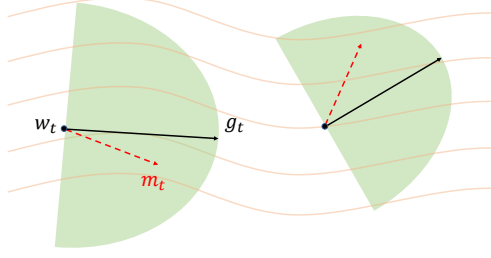


Figure 2: Assumption H3 on gradient prediction.

for a fixed batch of n samples $\{\xi_i\}_{i=1}^n$. The objective function $f(\cdot)$ is (potentially) nonconvex and has Lipschitz gradients. Set the terminating number, $T \in \{0, \dots, T_M - 1\}$, as a discrete r.v. with:

$$P(T = \ell) = \frac{\eta \ell}{\sum_{j=0}^{T_M-1} \eta_j}, \quad (4)$$

where T_M is the maximum number of iteration. The random termination number (4) is inspired by [Ghadimi and Lan, 2013] and is widely used to derive novel results in nonconvex optimization. Consider the following assumptions:

H1. For any $t > 0$, the estimated parameter w_t stays within a ℓ_∞ -ball. There exist a constant $W > 0$ such that $\|w_t\|_\infty \leq W$ almost surely.

H2. The function f is L -smooth (has L -Lipschitz gradients) w.r.t. the parameter w . There exist some constant $L > 0$ such that for $(w, \vartheta) \in \Theta^2$, $f(w) - f(\vartheta) - \nabla f(\vartheta)^\top (w - \vartheta) \leq \frac{L}{2} \|w - \vartheta\|^2$.

For nonconvex analysis, we assume the following:

H3. For any $t > 0$, $0 < \langle m_t, g_t \rangle = a_t \|g_t\|^2$ with some $0 < a_t \leq 1$, and $\|m_t\| \leq \|g_t\|$, where $\langle \cdot \rangle$ denotes the inner product.

H3 assumes that the predicted gradient is in general reasonable, in the sense that m_t has acute angle with g_t and bounded norm, as the shadowed area in Figure 2. Lastly, We make a classical assumption in nonconvex optimization on the magnitude of the gradient:

H4. There exist a constant $M > 0$ such that for any w and ξ , it holds that $\|\nabla f(w, \xi)\| < M$.

We now derive important results for our global analysis. The first one ensures bounded norms of quantities of interests (resulting from the bounded stochastic gradient assumption):

Lemma 1. Assume H4, then the quantities defined in Algorithm 2 satisfy for any $w \in \Theta$ and $t > 0$, $\|\nabla f(w_t)\| < M$, $\|\theta_t\| < M$ and $\|\hat{v}_t\| < M^2$.

We now formulate the main result of our paper yielding a finite-time upper bound of the suboptimality condition defined as $\mathbb{E} [\|\nabla f(w_T)\|^2]$ (set as the convergence criterion of interest, see [Ghadimi and Lan, 2013]):

Theorem 2. Assume H1-H4, $\beta_1 < \beta_2 \in [0, 1)$ and a sequence of decreasing stepsizes $\{\eta_t\}_{t>0}$, then the following result holds:

$$\mathbb{E} [\|\nabla f(w_T)\|_2^2] \leq \tilde{C}_1 \sqrt{\frac{d}{T_M}} + \tilde{C}_2 \frac{1}{T_M},$$

where T is a random termination number distributed according (4). The constants are defined as:

$$\begin{aligned} \tilde{C}_1 &= \frac{M \left[\frac{a_m(1-\beta_1)^2}{1-\beta_2} + 2L \frac{1}{1-\beta_2} + \Delta f + \frac{4L\beta_1^2(1+\beta_1^2)}{(1-\beta_1)(1-\beta_2)(1-\gamma)} \right]}{(1-a_m\beta_1) + (\beta_1 + a_m)}, \\ \tilde{C}_2 &= \frac{(a_m\beta_1^2 - 2a_m\beta_1 + \beta_1)M^2}{(1-\beta_1)((1-a_m\beta_1) + (\beta_1 + a_m))} \mathbb{E} [\|\hat{v}_0^{-1/2}\|], \end{aligned}$$

where $\Delta f = f(\bar{w}_1) - f(\bar{w}_{T_M+1})$ and $a_m = \min_{t=1, \dots, T} a_t$.

203 Firstly, the bound for our OPT-AMSGrad method matches the complexity bound of $\mathcal{O}(\sqrt{d/T_M} +$
 204 $1/T_M)$ of [Ghadimi and Lan, 2013] for SGD considering the dependence of T only, and of [Zhou
 205 et al., 2018] for AMSGrad method. To see the influence of prediction quality, we can show that when
 206 $(1 - \beta_1)(\beta_2 - \beta_1^2 - 2L(1 - \beta_1)) - \frac{4L\beta_1^2(1+\beta_1^2)}{1-\gamma} < 0$, \tilde{C}_1 and \tilde{C}_2 both decrease as a_m approaches
 207 1, i.e. as the prediction gets more accurate. Therefore, similar to the convex case, our bound also
 208 improves with better gradient prediction.

209 4.3 Checking H1 for a Deep Neural Network

210 As boundedness assumption H1 is generally hard to verify, we now show, for illustrative purposes,
 211 that the weights of a fully connected feed forward neural network stay in a bounded set when being
 212 trained using our method. The activation function for this section will be sigmoid function and we
 213 use a ℓ_2 regularization. We consider a fully connected feed forward neural network with L layers
 214 modeled by the function $\text{MLN}(w, \xi) : \Theta^d \times \mathbb{R}^p \rightarrow \mathbb{R}$ defined as:

$$\text{MLN}(w, \xi) = \sigma \left(w^{(L)} \sigma \left(w^{(L-1)} \dots \sigma \left(w^{(1)} \xi \right) \right) \right), \quad (5)$$

where $w = [w^{(1)}, w^{(2)}, \dots, w^{(L)}]$ is the vector of parameters, $\xi \in \mathbb{R}^p$ is the input data and σ is the
 sigmoid activation function. We assume a p dimension input data and a scalar output for simplicity.
 In this setting, the stochastic objective function (3) reads

$$f(w, \xi) = \mathcal{L}(\text{MLN}(w, \xi), y) + \frac{\lambda}{2} \|w\|^2,$$

215 where $\mathcal{L}(\cdot, y)$ is the loss function (e.g., cross-entropy), y are the true labels and $\lambda > 0$ is the regular-
 216 ization parameter. We establish that the boundedness assumption H1 is satisfied with model (5) via
 217 the following:

218 **Lemma 2.** *Given the multilayer model (5), assume the boundedness of the input data and of the*
 219 *loss function, i.e., for any $\xi \in \mathbb{R}^p$ and $y \in \mathbb{R}$ there is a constant $T > 0$ such that $\|\xi\| \leq 1$ a.s.*
 220 *and $|\mathcal{L}'(\cdot, y)| \leq T$ where $\mathcal{L}'(\cdot, y)$ denotes its derivative w.r.t. the parameter. Then for each layer*
 221 *$\ell \in [1, L]$, there exist a constant $A_{(\ell)}$ such that $\|w^{(\ell)}\| \leq A_{(\ell)}$*

222 5 Comparison to related methods

223 **Comparison to nonconvex optimization works.** Recently, [Zaheer et al., 2018, Chen et al.,
 224 2019a, Ward et al., 2019, Zhou et al., 2018, Zou and Shen, 2018, Li and Orabona., 2019] pro-
 225 vide some theoretical analysis of ADAM-type algorithms when applying them to smooth non-
 226 convex optimization problems. For example, [Chen et al., 2019a] provide the following bound
 227 $\min_{t \in [T]} \mathbb{E}[\|\nabla f(w_t)\|^2] = \mathcal{O}(\log T / \sqrt{T})$. Yet, this data independent bound does not show any ad-
 228 vantage over standard stochastic gradient descent. Similar concerns appear in other related works.
 229 To get some adaptive data dependent bound written in terms of the gradient norms observed along
 230 the trajectory when applying OPT-AMSGRAD to nonconvex optimization, one can follow the ap-
 231 proach of [Agarwal et al., 2019] or [Chen et al., 2019b]. They provide a modular approach to convert
 232 algorithms with adaptive data dependent regret bound for convex loss functions (e.g., ADAGRAD)
 233 to algorithms that can find an approximate stationary point of nonconvex objectives. These variants
 234 can outperform the ones instantiated by other ADAM-type algorithms when the gradient prediction
 235 m_t is close to the true gradient g_t .

236 **Comparison to AO-FTRL [Mohri and Yang, 2016].** In [Mohri and Yang, 2016], the authors
 237 propose AO-FTRL, which update reads $w_{t+1} = \arg \min_{w \in \Theta} (\sum_{s=1}^t g_s)^\top w + m_{t+1}^\top w + r_{0:t}(w)$,
 238 where $r_{0:t}(\cdot)$ is a 1-strongly convex loss function with respect to some norm $\|\cdot\|_{(t)}$ that may be
 239 different for different iteration t . Data dependent regret bound provided in [Mohri and Yang, 2016]
 240 reads $r_{0:T}(w^*) + \sum_{t=1}^T \|g_t - m_t\|_{(t)}^*$ for any benchmark $w^* \in \Theta$. We remark that if one selects
 241 $r_{0:t}(w) := \langle w, \text{diag}\{\hat{v}_t\}^{1/2} w \rangle$ and $\|\cdot\|_{(t)} := \sqrt{\langle \cdot, \text{diag}\{\hat{v}_t\}^{1/2} \cdot \rangle}$, then the update might be viewed
 242 as an optimistic variant of ADAGRAD. However, no experiments were provided in [Mohri and Yang,
 243 2016] to back those findings.

244 **Comparison to OPTIMISTIC-ADAM [Daskalakis et al., 2018].** This is an optimistic variant of
 245 ADAM, namely OPTIMISTIC-ADAM. A slightly modified version is summarized in Algorithm 3.

Algorithm 3 OPTIMISTIC-ADAM [DASKALAKIS ET AL., 2018]+ \hat{v}_t .

```

1: Required: parameter  $\beta_1, \beta_2$ , and  $\eta_t$ .
2: Init:  $w_1 \in \Theta$  and  $\hat{v}_0 = v_0 = \epsilon \mathbf{1} \in \mathbb{R}^d$ .
3: for  $t = 1$  to  $T$  do
4:   Get mini-batch stochastic gradient vector  $g_t \in \mathbb{R}^d$  at  $w_t$ .
5:    $\theta_t = \beta_1 \theta_{t-1} + (1 - \beta_1) g_t$ .
6:    $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$ .
7:    $\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$ .
8:    $w_{t+1} = \Pi_k[w_t - 2\eta_t \frac{\theta_t}{\sqrt{\hat{v}_t}} + \eta_t \frac{\theta_{t-1}}{\sqrt{\hat{v}_{t-1}}}]$ .
9: end for

```

246 Here, OPTIMISTIC-ADAM+ \hat{v}_t corresponds to OPTIMISTIC-ADAM with the additional max opera-
 247 tion $\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$ to guarantee that the weighted second moment is monotone increasing.

248 We want to emphasize that the motivations of our optimistic algorithm are different. OPTIMISTIC-
 249 ADAM is designed to optimize two-player games (e.g., GANs [Goodfellow et al., 2014]), while our
 250 proposed algorithm OPT-AMSGRAD is designed to accelerate optimization (e.g., solving empirical
 251 risk minimization). [Daskalakis et al., 2018] focuses on training GANs [Goodfellow et al., 2014] as
 252 a two-player zero-sum game. [Daskalakis et al., 2018] is inspired by these related works and shows
 253 that OPTIMISTIC-MIRROR-DESCENT can avoid the cycle behavior in a bilinear zero-sum game thus
 254 accelerating convergence.

255 6 Numerical Experiments

256 In this section, we provide experiments on classification tasks with various neural network archi-
 257 tectures and datasets to demonstrate the effectiveness of OPT-AMSGRAD in practice and justify
 258 its theoretical convergence acceleration. We start with giving an overview of the gradient predictor
 259 process before presenting our numerical runs.

260 6.1 Gradient Estimation

261 Based on the analysis in the previous section, we understand that the choice of the prediction m_t
 262 plays an important role in the convergence of OPTIMISTIC-AMSGRAD. Some classical works in
 263 gradient prediction methods include ANDERSON acceleration [Walker and Ni., 2011], MINIMAL
 264 POLYNOMIAL EXTRAPOLATION [Cabay and Jackson, 1976] and REDUCED RANK EXTRAPOLA-
 265 TION [Eddy, 1979]. These methods aim at finding a fixed point g^* and assume that $\{g_t \in \mathbb{R}^d\}_{t>0}$
 266 has the following linear relation:

$$g_t - g^* = A(g_{t-1} - g^*) + e_t, \quad (6)$$

267 where e_t is a second order term satisfying $\|e_t\|_2 = \mathcal{O}(\|g_{t-1} - g^*\|_2^2)$ and $A \in \mathbb{R}^{d \times d}$ is an unknown
 268 matrix, see [Scieur et al., 2016] for details and results. For our numerical experiments, we run OPT-
 269 AMSGRAD using Algorithm 4 to construct the sequence $\{m_t\}_{t>0}$ which is based on estimating the
 270 limit of a sequence using the last iterates [Brezinski and Zaglia, 2013].

Algorithm 4 Regularized Approximated Minimal Polynomial Extrapolation [Scieur et al., 2016]

```

1: Input: sequence  $\{g_s \in \mathbb{R}^d\}_{s=0}^{s=r-1}$ , parameter  $\lambda > 0$ .
2: Compute matrix  $U = [g_1 - g_0, \dots, g_r - g_{r-1}] \in \mathbb{R}^{d \times r}$ .
3: Obtain  $z$  by solving  $(U^\top U + \lambda I)z = \mathbf{1}$ .
4: Get  $c = z/(z^\top \mathbf{1})$ .
5: Output:  $\sum_{i=0}^{r-1} c_i g_i$ , the approximation of the fixed point  $g^*$ .

```

271 Specifically, at iteration t , m_t is obtained by (a) calling Algorithm 4 with a sequence of r past
 272 gradients, $\{g_{t-1}, g_{t-2}, \dots, g_{t-r}\}$ as input yielding the vector $c = [c_0, \dots, c_{r-1}]$ and (b) setting
 273 $m_t := \sum_{i=0}^{r-1} c_i g_{t-r+i}$. To understand why the output from the extrapolation method may be a
 274 reasonable estimation, assume that the update converges to a stationary point (i.e. $g^* := \nabla f(w^*) =$

0 for the underlying function f). Then, we might rewrite (6) as $g_t = Ag_{t-1} + \mathcal{O}(\|g_{t-1}\|_2^2)u_{t-1}$, for some unit vector u_{t-1} . This equation suggests that the next gradient vector g_t is a linear transform of g_{t-1} plus an error vector that may not be in the span of A . If the algorithm converges to a stationary point, the magnitude of the error will converge to zero. We note that prior known gradient prediction methods are mainly designed for convex functions. Algorithm 4 is employed in our following numerical applications given its empirical success in Deep Learning, see Scieur et al. [2018], nevertheless, any gradient prediction method can be embedded in our OPT-AMSGRAD framework. The search for the optimal prediction process in order to accelerate even more OPT-AMSGRAD is an interesting research direction, which is left as future work.

Computational cost: This extrapolation step consists in: (a) Constructing the linear system $(U^\top U)$ which cost can be optimized to $\mathcal{O}(d)$, since the matrix U only changes one column at a time. (b) Solving the linear system which cost is $\mathcal{O}(r^3)$, and is negligible for a small r used in practice. (c) Outputting a weighted average of previous gradients which cost is $\mathcal{O}(r \times d)$ yielding a computational overhead of $\mathcal{O}((r+1)d + r^3)$. Yet, steps (a) and (c) can be parallelized in the final implementation.

6.2 Classification Experiments

Methods. We consider two baselines. The first one is the original AMSGRAD. The hyperparameters are set to be $\beta_1 = 0.9$ and $\beta_2 = 0.999$, see [Reddi et al., 2018]. The other benchmark method is the OPTIMISTIC-ADAM+ \hat{v}_t [Daskalakis et al., 2018], which described Algorithm 3. We use cross-entropy loss, a mini-batch size of 128 and tune the learning rates over a fine grid and report the best result for all methods. For OPT-AMSGRAD, we use $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and the best step size η of AMSGRAD for a fair evaluation of the optimistic step. In our implementation, OPT-AMSGRAD has an additional parameter r that controls the number of previous gradients used for gradient prediction. We use $r = 5$ past gradient for empirical reasons, see Section 6.3. The algorithms are initialized at the same point and the results are averaged over 5 repetitions.

Datasets. Following Reddi et al. [2018] and Kingma and Ba [2015], we compare different algorithms on *MNIST*, *CIFAR10*, *CIFAR100*, and *IMDB* datasets. For *MNIST*, we use two noisy variants namely *MNIST-back-rand* and *MNIST-back-image* from Larochelle et al. [2007]. They both have 12 000 training samples and 50 000 test samples, where random background is inserted to the original *MNIST* hand-written digit images. For *MNIST-back-rand*, each image is inserted with a random background, which pixel values are generated uniformly from 0 to 255, while *MNIST-back-image* takes random patches from a black and white noisy background. The input dimension is 784 (28×28) and the number of classes is 10. *CIFAR10* and *CIFAR100* are popular computer-vision datasets of 50 000 training images and 10 000 test images, of size 32×32 . The *IMDB* movie review dataset, popular for text classification, is a binary dataset with 25 000 training and testing samples respectively.

Network architectures. We adopt a multi-layer fully connected neural network with hidden layers of 200 connected to another layer with 100 neurons (using ReLU activations and Softmax output). This network is tested on *MNIST* variants. For convolutional networks, we adopt a simple four layer CNN which has 2 convolutional layers following by a fully connected layer. In addition, we also apply residual networks, Resnet-18 and Resnet-50 [He et al., 2016], which have achieved state-of-the-art results. For the texture *IMDB* dataset, we consider a Long-Short Term Memory (LSTM) network [Gers et al., 2000]. The latter network includes a word embedding layer with 5 000 input entries representing most frequent words embedded into a 32 dimensional space. The output of the embedding layer is passed to 100 LSTM units then connected to 100 fully connected ReLU layers.

Results. Firstly, to illustrate the acceleration effect of OPT-AMSGRAD at early stage, we provide the training loss against number of iterations in Figure 3. We clearly observe that on all datasets, the proposed OPT-AMSGRAD converges faster than the other competing methods since fewer iterations are required to achieve the same precision, validating one of the main edges of OPT-AMSGRAD. We are also curious about the long-term performance and generalization of the proposed method in test phase. In Figure 4, we plot the results when the model is trained until the test accuracy stabilizes. We observe: (1) in the long term, OPT-AMSGRAD algorithm may converge to a better point with smaller loss value, and (2) in these applications, our proposed OPT-AMSGRAD also outperforms the competing methods in terms of test accuracy.

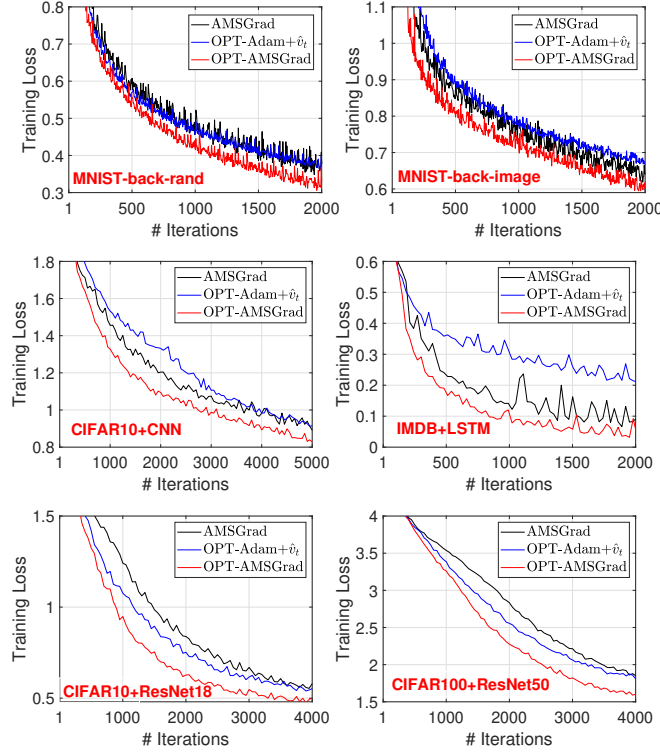


Figure 3: Training loss vs. Number of iterations for fully connected NN, CNN, LSTM and ResNet.

6.3 Choice of parameter r

Since the number of past gradients r is important in gradient prediction (Algorithm 4), we compare Figure 5 the performance under different values $r = 3, 5, 10$ on two datasets. From the results we see that, taking into consideration both quality of gradient prediction and computational cost, $r = 5$ is a good choice for most applications. We remark that, empirically, the performance comparison among $r = 3, 5, 10$ is not absolutely consistent (i.e. more means better) in all cases. We suspect one possible reason is that for deep neural networks, the diversity of computed gradients through the iterations, due to the highly nonconvex loss, makes them inefficient for sequentially building the predictable process $\{m_t\}_{t \geq 0}$. Thus, sometimes, the recent gradient vectors (e.g. $r \leq 5$) can be more informative. Yet, in some sense, this characteristic, very specific to deep neural networks, is itself a fundamental problem of gradient prediction methods.

7 Conclusion

In this paper, we propose OPT-AMSGRAD, which combines optimistic online learning and AMSGRAD to improve sample efficiency and accelerate the training process, in particular for fitting deep neural networks on a finite batch of observations. Given a well-designed gradient prediction process, we theoretically show that the regret, through the iterations, can be smaller than that of standard AMSGRAD. We also establish a finite-time convergence bound on the second order moment of the gradient of the objective loss function matching that of state-of-the-art adaptive gradient methods. Experiments on several benchmark datasets using various deep learning models demonstrate the effectiveness of the proposed algorithm in accelerating the empirical risk minimization procedure and empirically show better generalization properties of our method OPT-AMSGRAD.

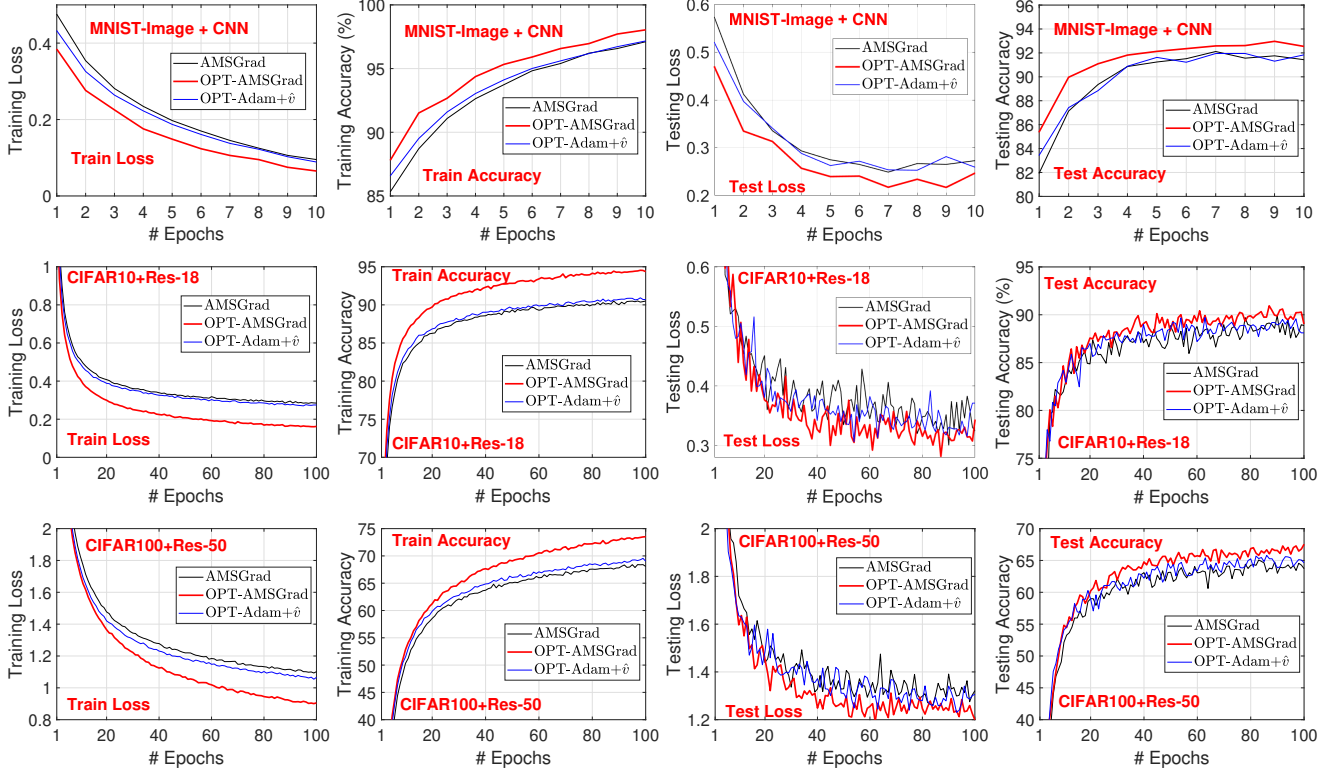


Figure 4: *MNIST-back-image + CNN*, *CIFAR10 + Res-18* and *CIFAR100 + Res-50*. We compare three methods in terms of training (cross-entropy) loss and accuracy, testing loss and accuracy.

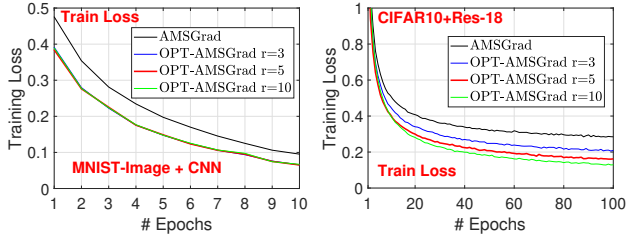


Figure 5: Training loss w.r.t. different r values.

References

- Jacob Abernethy, Kevin A. Lai, Kfir Y. Levy, and Jun-Kun Wang. Faster rates for convex-concave games. *COLT*, 2018.
- Naman Agarwal, Brian Bullins, Xinyi Chen, Elad Hazan, Karan Singh, Cyril Zhang, and Yi Zhang. Efficient full-matrix adaptive regularization. *ICML*, 2019.
- C. Brezinski and M. R. Zaglia. Extrapolation methods: theory and practice. *Elsevier*, 2013.
- S. Cabay and L. Jackson. A polynomial extrapolation method for finding limits and antilimits of vector sequences. *SIAM Journal on Numerical Analysis*, 1976.
- Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. *ICLR*, 2019a.
- Zaiyi Chen, Zhuoning Yuan, Jinfeng Yi, Bowen Zhou, Enhong Chen, and Tianbao Yang. Universal stagewise learning for non-convex problems with convergence on averaged solutions. *ICLR*, 2019b.
- Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. *COLT*, 2012.
- Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism. *ICLR*, 2018.
- Alexandre Défossez, Léon Bottou, Francis Bach, and Nicolas Usunier. On the convergence of adam and adagrad. *arXiv preprint arXiv:2003.02395*, 2020.
- Timothy Dozat. Incorporating nesterov momentum into adam. *ICLR (Workshop Track)*, 2016.
- John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research (JMLR)*, 2011.
- R. Eddy. Extrapolating to the limit of a vector sequence. *Information linkage between applied mathematics and industry*, Elsevier, 1979.
- Felix A. Gers, Jürgen A. Schmidhuber, and Fred A. Cummins. Learning to forget: Continual prediction with lstm. *Neural Comput.*, 12(10):2451–2471, October 2000. ISSN 0899-7667.
- Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NIPS*, 2014.
- Alex Graves, Abdel rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. *ICASSP*, 2013.
- Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2016.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2016.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.
- Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. *ICML*, 2007.
- Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *NIPS*, 2017.

391 Xiaoyu Li and Francesco Orabona. On the convergence of stochastic gradient descent with adaptive
392 stepsizes. *AISTAT*, 2019.

393 H. Brendan McMahan and Matthew J. Streeter. Adaptive bound optimization for online convex
394 optimization. *COLT*, 2010.

395 Panayotis Mertikopoulos, Bruno Lecouat, Houssam Zenati, Chuan-Sheng Foo, Vijay Chan-
396 drasekhar, and Georgios Piliouras. Optimistic mirror descent in saddle-point problems: Going
397 the extra (gradient) mile. *arXiv preprint arXiv:1807.02629*, 2018.

398 Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wier-
399 stra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *NIPS (Deep Learning
400 Workshop)*, 2013.

401 Mehryar Mohri and Scott Yang. Accelerating optimization via adaptive prediction. *AISTATS*, 2016.

402 Yurii Nesterov. Introductory lectures on convex optimization: A basic course. *Springer*, 2004.

403 B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *Mathematics and
404 Mathematical Physics*, 1964.

405 Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable se-
406 quences. In *Advances in Neural Information Processing Systems*, pages 3066–3074, 2013.

407 Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *ICLR*,
408 2018.

409 Damien Scieur, Alexandre d’Aspremont, and Francis Bach. Regularized nonlinear acceleration.
410 *NIPS*, 2016.

411 Damien Scieur, Edouard Oyallon, Alexandre d’Aspremont, and Francis Bach. Nonlinear accelera-
412 tion of deep neural networks. *CoRR*, abs/1805.09639, 2018.

413 Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E. Schapire. Fast convergence of
414 regularized learning in games. *NIPS*, 2015.

415 T. Tieleman and G. Hinton. Rmsprop: Divide the gradient by a running average of its recent magni-
416 tude. *COURSERA: Neural Networks for Machine Learning*, 2012.

417 Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. 2008.

418 H. F. Walker and P. Ni. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical
419 Analysis*, 2011.

420 Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over nonconvex
421 landscapes, from any initialization. *ICML*, 2019.

422 Yan Yan, Tianbao Yang, Zhe Li, Qihang Lin, and Yi Yang. A unified analysis of stochastic momen-
423 tum methods for deep learning. *arXiv preprint arXiv:1808.10396*, 2018.

424 Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive meth-
425 ods for nonconvex optimization. *NeurIPS*, 2018.

426 Matthew D. Zeiler. Adadelta: An adaptive learning rate method. *arXiv:1212.5701*, 2012.

427 Dongruo Zhou, Yiqi Tang, Ziyang Yang, Yuan Cao, and Quanquan Gu. On the convergence of
428 adaptive gradient methods for nonconvex optimization. *arXiv:1808.05671*, 2018.

429 Fangyu Zou and Li Shen. On the convergence of adagrad with momentum for training deep neural
430 networks. *arXiv:1808.03408*, 2018.

431 Appendix: An Optimistic Acceleration of AMSGrad for Noncon- 432 vex Optimization

433 A Proof of Theorem 1

434 **Theorem.** Suppose the learner incurs a sequence of convex loss functions $\{\ell_t(\cdot)\}$. Then, OPT-
435 AMSGRAD (Algorithm 2) has regret

$$\mathcal{R}_T \leq \frac{B_{\psi_1}(w^*, \tilde{w}_1)}{\eta_1} + \sum_{t=1}^T \frac{\eta_t}{2} \|g_t - \tilde{m}_t\|_{\psi_{t-1}^*}^2 + \frac{D_\infty^2}{\eta_{\min}} \sum_{i=1}^d \hat{v}_T^{1/2}[i] + D_\infty^2 \beta_1^2 \sum_{t=1}^T \|g_t - \theta_{t-1}\|_{\psi_{t-1}^*},$$

436 where $\tilde{m}_{t+1} = \beta_1 \theta_{t-1} + (1 - \beta_1) m_{t+1}$, $g_t := \nabla \ell_t(w_t)$, $\eta_{\min} := \min_t \eta_t$ and D_∞^2 is the diameter of
437 the bounded set Θ . The result holds for any benchmark $w^* \in \Theta$ and any step size sequence $\{\eta_t\}_{t>0}$.

438 **Proof** Beforehand, we denote:

$$\begin{aligned} \tilde{g}_t &= \beta_1 \theta_{t-1} + (1 - \beta_1) g_t, \\ \tilde{m}_{t+1} &= \beta_1 \theta_{t-1} + (1 - \beta_1) m_{t+1}, \end{aligned} \quad (7)$$

439 where we recall that g_t and m_{t+1} are respectively the gradient $\nabla \ell_t(w_t)$ and the predictable guess.
440 By regret decomposition, we have that

$$\begin{aligned} \mathcal{R}_T &:= \sum_{t=1}^T \ell_t(w_t) - \min_{w \in \Theta} \sum_{t=1}^T \ell_t(w) \\ &\leq \sum_{t=1}^T \langle w_t - w^*, \nabla \ell_t(w_t) \rangle \\ &= \sum_{t=1}^T \langle w_t - \tilde{w}_{t+1}, g_t - \tilde{m}_t \rangle + \langle w_t - \tilde{w}_{t+1}, \tilde{m}_t \rangle + \langle \tilde{w}_{t+1} - w^*, \tilde{g}_t \rangle + \langle \tilde{w}_{t+1} - w^*, g_t - \tilde{g}_t \rangle. \end{aligned} \quad (8)$$

441 Recall the notation $\psi_t(x)$ and the Bregman divergence $B_{\psi_t}(u, v)$ defined Section 4. We ex-
442 ploit a useful inequality (which appears in e.g., [Tseng, 2008]). For any update of the form
443 $\hat{w} = \arg \min_{w \in \Theta} \langle w, \theta \rangle + B_{\psi_t}(w, v)$, it holds that

$$\langle \hat{w} - u, \theta \rangle \leq B_{\psi_t}(u, v) - B_{\psi_t}(u, \hat{w}) - B_{\psi_t}(\hat{w}, v) \quad \text{for any } u \in \Theta. \quad (9)$$

444 For $\beta_1 = 0$, we can rewrite the update on line 8 of (Algorithm 2) as

$$\tilde{w}_{t+1} = \arg \min_{w \in \Theta} \eta_t \langle w, \tilde{g}_t \rangle + B_{\psi_t}(w, \tilde{w}_t). \quad (10)$$

445 By using (9) for (10) with $\hat{w} = \tilde{w}_{t+1}$ (the output of the minimization problem), $u = w^*$ and $v = \tilde{w}_t$,
446 we have

$$\langle \tilde{w}_{t+1} - w^*, \tilde{g}_t \rangle \leq \frac{1}{\eta_t} [B_{\psi_t}(w^*, \tilde{w}_t) - B_{\psi_t}(w^*, \tilde{w}_{t+1}) - B_{\psi_t}(\tilde{w}_{t+1}, \tilde{w}_t)]. \quad (11)$$

447 We can also rewrite the update on line 9 of (Algorithm 2) at time t as

$$w_{t+1} = \arg \min_{w \in \Theta} \eta_{t+1} \langle w, \tilde{m}_{t+1} \rangle + B_{\psi_t}(w, \tilde{w}_{t+1}). \quad (12)$$

448 and, by using (9) for (12) (written at iteration t), with $\hat{w} = w_t$ (the output of the minimization
449 problem), $u = \tilde{w}_{t+1}$ and $v = \tilde{w}_t$, we have

$$\langle w_t - \tilde{w}_{t+1}, \tilde{m}_t \rangle \leq \frac{1}{\eta_t} [B_{\psi_{t-1}}(\tilde{w}_{t+1}, \tilde{w}_t) - B_{\psi_{t-1}}(\tilde{w}_{t+1}, w_t) - B_{\psi_{t-1}}(w_t, \tilde{w}_t)]. \quad (13)$$

By (8), (11), and (13), we obtain

$$\begin{aligned}
\mathcal{R}_T &\stackrel{(8)}{\leq} \sum_{t=1}^T \langle w_t - \tilde{w}_{t+1}, g_t - \tilde{m}_t \rangle + \langle w_t - \tilde{w}_{t+1}, \tilde{m}_t \rangle + \langle \tilde{w}_{t+1} - w^*, \tilde{g}_t \rangle + \langle \tilde{w}_{t+1} - w^*, g_t - \tilde{g}_t \rangle \\
&\stackrel{(11),(13)}{\leq} \sum_{t=1}^T \|w_t - \tilde{w}_{t+1}\|_{\psi_{t-1}} \|g_t - \tilde{m}_t\|_{\psi_{t-1}^*} + \|\tilde{w}_{t+1} - w^*\|_{\psi_{t-1}} \|g_t - \tilde{g}_t\|_{\psi_{t-1}^*} \\
&\quad + \frac{1}{\eta_t} [B_{\psi_{t-1}}(\tilde{w}_{t+1}, \tilde{w}_t) - B_{\psi_{t-1}}(\tilde{w}_{t+1}, w_t) - B_{\psi_{t-1}}(w_t, \tilde{w}_t) \\
&\quad + B_{\psi_t}(w^*, \tilde{w}_t) - B_{\psi_t}(w^*, \tilde{w}_{t+1}) - B_{\psi_t}(\tilde{w}_{t+1}, \tilde{w}_t)] ,
\end{aligned} \tag{14}$$

which is further bounded by

$$\begin{aligned}
\mathcal{R}_T &\leq \sum_{t=1}^T \left\{ \frac{1}{2\eta_t} \|w_t - \tilde{w}_{t+1}\|_{\psi_{t-1}}^2 + \frac{\eta_t}{2} \|g_t - m_t\|_{\psi_{t-1}^*}^2 + \|\tilde{w}_{t+1} - w^*\|_{\psi_{t-1}} \|g_t - \tilde{g}_t\|_{\psi_{t-1}^*} \right. \\
&\quad \left. + \frac{1}{\eta_t} \underbrace{(B_{\psi_{t-1}}(\tilde{w}_{t+1}, \tilde{w}_t) - B_{\psi_t}(\tilde{w}_{t+1}, \tilde{w}_t))}_{A_1} - \frac{1}{2} \|\tilde{w}_{t+1} - w_t\|_{\psi_{t-1}}^2 \right. \\
&\quad \left. + \underbrace{B_{\psi_t}(w^*, \tilde{w}_t) - B_{\psi_t}(w^*, \tilde{w}_{t+1}))}_{A_2} \right\} ,
\end{aligned} \tag{15}$$

where the inequality is due to $\|w_t - \tilde{w}_{t+1}\|_{\psi_{t-1}} \|g_t - m_t\|_{\psi_{t-1}^*} = \inf_{\beta > 0} \frac{1}{2\beta} \|w_t - \tilde{w}_{t+1}\|_{\psi_{t-1}}^2 + \frac{\beta}{2} \|g_t - m_t\|_{\psi_{t-1}^*}^2$ by Young's inequality and the 1-strongly convex of $\psi_{t-1}(\cdot)$ with respect to $\|\cdot\|_{\psi_{t-1}}$ which yields that $B_{\psi_{t-1}}(\tilde{w}_{t+1}, w_t) \geq \frac{1}{2} \|\tilde{w}_{t+1} - w_t\|_{\psi_t}^2 \geq 0$.

To proceed, notice that

$$\begin{aligned}
A_1 &:= B_{\psi_{t-1}}(\tilde{w}_{t+1}, \tilde{w}_t) - B_{\psi_t}(\tilde{w}_{t+1}, \tilde{w}_t) \\
&= \langle \tilde{w}_{t+1} - \tilde{w}_t, \text{diag}(\hat{v}_{t-1}^{1/2} - \hat{v}_t^{1/2})(\tilde{w}_{t+1} - \tilde{w}_t) \rangle \leq 0 ,
\end{aligned} \tag{16}$$

as the sequence $\{\hat{v}_t\}$ is non-decreasing. And that

$$\begin{aligned}
A_2 &:= B_{\psi_t}(w^*, \tilde{w}_t) - B_{\psi_t}(w^*, \tilde{w}_{t+1}) = \langle w^* - \tilde{w}_{t+1}, \text{diag}(\hat{v}_{t+1}^{1/2} - \hat{v}_t^{1/2})(w^* - \tilde{w}_{t+1}) \rangle \\
&\leq (\max_i (w^*[i] - \tilde{w}_{t+1}[i])^2) \cdot \left(\sum_{i=1}^d \hat{v}_{t+1}^{1/2}[i] - \hat{v}_t^{1/2}[i] \right) .
\end{aligned} \tag{17}$$

Therefore, by (15), (17), (16), we have

$$\mathcal{R}_T \leq \frac{D_\infty^2}{\eta_{\min}} \sum_{i=1}^d \hat{v}_T^{1/2}[i] + \frac{B_{\psi_1}(w^*, \tilde{w}_1)}{\eta_1} + \sum_{t=1}^T \frac{\eta_t}{2} \|g_t - \tilde{m}_t\|_{\psi_{t-1}^*}^2 + D_\infty^2 \beta_1^2 \sum_{t=1}^T \|g_t - \theta_{t-1}\|_{\psi_{t-1}^*} ,$$

since $\|g_t - \tilde{g}_t\|_{\psi_{t-1}^*} = \|g_t - \beta_1 \theta_{t-1} - (1 - \beta_1) g_t\|_{\psi_{t-1}^*} = \beta^2 \|g_t - \theta_{t-1}\|_{\psi_{t-1}^*}$. This completes the proof.

□

B Proof of Corollary 1

Corollary. Suppose $\beta_1 = 0$ and $\{v_t\}_{t>0}$ is a monotonically increasing sequence, then we obtain the following regret bound for any $w^* \in \Theta$ and sequence of stepsizes $\{\eta_t = \eta/\sqrt{t}\}_{t>0}$:

$$\mathcal{R}_T \leq \frac{B_{\psi_1}}{\eta_1} + \frac{\eta \sqrt{1 + \log T}}{\sqrt{1 - \beta_2}} \sum_{i=1}^d \|(g - m)_{1:T}[i]\|_2 + \frac{D_\infty^2}{\eta_{\min}} \sum_{i=1}^d \left[(1 - \beta_2) \sum_{s=1}^T \beta_2^{T-s} g_s^2[i] \right]^{1/2} ,$$

where $B_{\psi_1} := B_{\psi_1}(w^*, \tilde{w}_1)$, $g_t := \nabla \ell_t(w_t)$ and $\eta_{\min} := \min_t \eta_t$.

465 **Proof** Recall the bound in Theorem 1:

$$\mathcal{R}_T \leq \frac{B_{\psi_1}(w^*, \tilde{w}_1)}{\eta_1} + \sum_{t=1}^T \frac{\eta_t}{2} \|g_t - \tilde{m}_t\|_{\psi_{t-1}^*}^2 + \frac{D_\infty^2}{\eta_{\min}} \sum_{i=1}^d \hat{v}_T^{1/2}[i] + D_\infty^2 \beta_1^2 \sum_{t=1}^T \|g_t - \theta_{t-1}\|_{\psi_{t-1}^*}.$$

466 The second term reads:

$$\begin{aligned} & \sum_{t=1}^T \frac{\eta_t}{2} \|g_t - m_t\|_{\psi_{t-1}^*}^2 \\ &= \sum_{t=1}^{T-1} \frac{\eta_t}{2} \|g_t - m_t\|_{\psi_{t-1}^*}^2 + \eta_T \sum_{i=1}^d \frac{(g_T[i] - m_T[i])^2}{\sqrt{v_{T-1}[i]}} \\ &= \sum_{t=1}^{T-1} \frac{\eta_t}{2} \|g_t - m_t\|_{\psi_{t-1}^*}^2 + \eta \sum_{i=1}^d \frac{(g_T[i] - m_T[i])^2}{\sqrt{T((1-\beta_2) \sum_{s=1}^{T-1} \beta_2^{T-1-s} (g_s[i] - m_s[i])^2)}} \\ &\leq \eta \sum_{i=1}^d \sum_{t=1}^T \frac{(g_t[i] - m_t[i])^2}{\sqrt{t((1-\beta_2) \sum_{s=1}^{t-1} \beta_2^{t-1-s} (g_s[i] - m_s[i])^2)}}. \end{aligned}$$

467 To interpret the bound, let us make a rough approximation such that $\sum_{s=1}^{t-1} \beta_2^{t-1-s} (g_s[i] - m_s[i])^2 \simeq$
 468 $(g_t[i] - m_t[i])^2$. Then, we can further get an upper-bound as

$$\sum_{t=1}^T \frac{\eta_t}{2} \|g_t - m_t\|_{\psi_{t-1}^*}^2 \leq \frac{\eta}{\sqrt{1-\beta_2}} \sum_{i=1}^d \sum_{t=1}^T \frac{|g_t[i] - m_t[i]|}{\sqrt{t}} \leq \frac{\eta \sqrt{1+\log T}}{\sqrt{1-\beta_2}} \sum_{i=1}^d \|(g - m)_{1:T}[i]\|_2,$$

469 where the last inequality is due to Cauchy-Schwarz.

470 □

471 C Proofs of Auxiliary Lemmas

472 Following [Yan et al., 2018] and their study of the SGD with Momentum we denote for any $t > 0$:

$$\bar{w}_t = w_t + \frac{\beta_1}{1-\beta_1} (w_t - \tilde{w}_{t-1}) = \frac{1}{1-\beta_1} w_t - \frac{\beta_1}{1-\beta_1} \tilde{w}_{t-1}. \quad (18)$$

473 **Lemma 3.** Assume a strictly positive and non increasing sequence of stepsizes $\{\eta_t\}_{t>0}$, $\beta_1 < \beta_2 \in$
 474 $[0, 1)$, then the following holds:

$$\bar{w}_{t+1} - \bar{w}_t \leq \frac{\beta_1}{1-\beta_1} \tilde{\theta}_{t-1} \left[\eta_{t-1} \hat{v}_{t-1}^{-1/2} - \eta_t \hat{v}_t^{-1/2} \right] - \eta_t \hat{v}_t^{-1/2} \tilde{g}_t,$$

475 where $\tilde{\theta}_t = \theta_t + \beta_1 \theta_{t-1}$ and $\tilde{g}_t = g_t - \beta_1 m_t + \beta_1 g_{t-1} + m_{t+1}$.

476 **Proof** By definition (18) and using the Algorithm updates, we have:

$$\begin{aligned} \bar{w}_{t+1} - \bar{w}_t &= \frac{1}{1-\beta_1} (w_{t+1} - \tilde{w}_t) - \frac{\beta_1}{1-\beta_1} (w_t - \tilde{w}_{t-1}) \\ &= -\frac{1}{1-\beta_1} \eta_t \hat{v}_t^{-1/2} (\theta_t + h_{t+1}) + \frac{\beta_1}{1-\beta_1} \eta_{t-1} \hat{v}_{t-1}^{-1/2} (\theta_{t-1} + h_t) \\ &= -\frac{1}{1-\beta_1} \eta_t \hat{v}_t^{-1/2} (\theta_t + \beta_1 \theta_{t-1}) - \frac{1}{1-\beta_1} \eta_t \hat{v}_t^{-1/2} (1-\beta_1) m_{t+1} \\ &\quad + \frac{\beta_1}{1-\beta_1} \eta_{t-1} \hat{v}_{t-1}^{-1/2} (\theta_{t-1} + \beta_1 \theta_{t-2}) + \frac{\beta_1}{1-\beta_1} \eta_{t-1} \hat{v}_{t-1}^{-1/2} (1-\beta_1) m_t. \end{aligned} \quad (19)$$

477 Denote $\tilde{\theta}_t = \theta_t + \beta_1 \theta_{t-1}$ and $\tilde{g}_t = g_t - \beta_1 m_t + \beta_1 g_{t-1} + m_{t+1}$. Notice that $\tilde{\theta}_t = \beta_1 \tilde{\theta}_{t-1} + (1 -$
 478 $\beta_1)(g_t + \beta_1 g_{t-1})$.

$$\bar{w}_{t+1} - \bar{w}_t \leq \frac{\beta_1}{1 - \beta_1} \tilde{\theta}_{t-1} \left[\eta_{t-1} \hat{v}_{t-1}^{-1/2} - \eta_t \hat{v}_t^{-1/2} \right] - \eta_t \hat{v}_t^{-1/2} \tilde{g}_t. \quad (20)$$

479 □

480 **Lemma 4.** Assume H4, a strictly positive and a sequence of constant stepsizes $\{\eta_t\}_{t>0}$, $(\beta_1, \beta_2) \in$
 481 $[0, 1]$, then the following holds:

$$\sum_{t=1}^{T_M} \eta_t^2 \mathbb{E} \left[\left\| \hat{v}_t^{-1/2} \theta_t \right\|_2^2 \right] \leq \frac{\eta^2 d T_M (1 - \beta_1)}{(1 - \beta_2)(1 - \gamma)}. \quad (21)$$

482 **Proof** We denote by index $p \in [1, d]$ the dimension of each component of vectors of interest. Noting
 483 that for any $t > 0$ and dimension p we have $\hat{v}_{t,p} \geq v_{t,p}$, then:

$$\begin{aligned} \eta_t^2 \mathbb{E} \left[\left\| \hat{v}_t^{-1/2} \theta_t \right\|_2^2 \right] &= \eta_t^2 \mathbb{E} \left[\sum_{p=1}^d \frac{\theta_{t,p}^2}{\hat{v}_{t,p}} \right] \\ &\leq \eta_t^2 \mathbb{E} \left[\sum_{i=1}^d \frac{\theta_{t,p}^2}{v_{t,p}} \right] \\ &\leq \eta_t^2 \mathbb{E} \left[\sum_{i=1}^d \frac{(\sum_{r=1}^t (1 - \beta_1) \beta_1^{t-r} g_{r,p})^2}{\sum_{r=1}^t (1 - \beta_2) \beta_2^{t-r} g_{r,p}^2} \right], \end{aligned} \quad (22)$$

484 where the last inequality is due to initializations. Denote $\gamma = \frac{\beta_1}{\beta_2}$. Then,

$$\begin{aligned} \eta_t^2 \mathbb{E} \left[\left\| \hat{v}_t^{-1/2} \theta_t \right\|_2^2 \right] &\leq \frac{\eta_t^2 (1 - \beta_1)^2}{1 - \beta_2} \mathbb{E} \left[\sum_{i=1}^d \frac{(\sum_{r=1}^t \beta_1^{t-r} g_{r,p})^2}{\sum_{r=1}^t \beta_2^{t-r} g_{r,p}^2} \right] \\ &\stackrel{(a)}{\leq} \frac{\eta_t^2 (1 - \beta_1)}{1 - \beta_2} \mathbb{E} \left[\sum_{i=1}^d \frac{\sum_{r=1}^t \beta_1^{t-r} g_{r,p}^2}{\sum_{r=1}^t \beta_2^{t-r} g_{r,p}^2} \right] \\ &\leq \frac{\eta_t^2 (1 - \beta_1)}{1 - \beta_2} \mathbb{E} \left[\sum_{i=1}^d \sum_{r=1}^t \gamma^{t-r} \right] = \frac{\eta_t^2 d (1 - \beta_1)}{1 - \beta_2} \mathbb{E} \left[\sum_{r=1}^t \gamma^{t-r} \right], \end{aligned} \quad (23)$$

485 where (a) is due to $\sum_{r=1}^t \beta_1^{t-r} \leq \frac{1}{1 - \beta_1}$. Summing from $t = 1$ to $t = T_M$ on both sides yields:

$$\begin{aligned} \sum_{t=1}^{T_M} \eta_t^2 \mathbb{E} \left[\left\| \hat{v}_t^{-1/2} \theta_t \right\|_2^2 \right] &\leq \frac{\eta^2 d (1 - \beta_1)}{1 - \beta_2} \mathbb{E} \left[\sum_{t=1}^{T_M} \sum_{r=1}^t \gamma^{t-r} \right] \\ &\leq \frac{\eta^2 d T (1 - \beta_1)}{1 - \beta_2} \mathbb{E} \left[\sum_{t=t}^t \gamma^{t-r} \right] \\ &\leq \frac{\eta^2 d T (1 - \beta_1)}{(1 - \beta_2)(1 - \gamma)}, \end{aligned} \quad (24)$$

486 where the last inequality is due to $\sum_{r=1}^t \gamma^{t-r} \leq \frac{1}{1 - \gamma}$ by definition of γ . □

487 C.1 Proof of Lemma 1

Lemma. Assume assumption H4, then the quantities defined in Algorithm 2 satisfy for any $w \in \Theta$
 and $t > 0$:

$$\|\nabla f(w_t)\| < M, \quad \|\theta_t\| < M, \quad \|\hat{v}_t\| < M^2.$$

Proof Assume assumption H4 we have:

$$\|\nabla f(w)\| = \|\mathbb{E}[\nabla f(w, \xi)]\| \leq \mathbb{E}[\|\nabla f(w, \xi)\|] \leq M.$$

488 By induction reasoning, since $\|\theta_0\| = 0 \leq M$ and suppose that for $\|\theta_t\| \leq M$ then we have

$$\|\theta_{t+1}\| = \|\beta_1 \theta_t + (1 - \beta_1) g_{t+1}\| \leq \beta_1 \|\theta_t\| + (1 - \beta_1) \|g_{t+1}\| \leq M. \quad (25)$$

489 Using the same induction reasoning we prove that

$$\|\hat{v}_{t+1}\| = \|\beta_2 \hat{v}_t + (1 - \beta_2) g_{t+1}^2\| \leq \beta_2 \|\hat{v}_t\| + (1 - \beta_1) \|g_{t+1}^2\| \leq M^2. \quad (26)$$

490

□

491 D Proof of Theorem 2

492 **Theorem.** Assume H1-H4, $\beta_1 < \beta_2 \in [0, 1)$ and a sequence of decreasing stepsizes $\{\eta_t\}_{t>0}$, then
493 the following result holds:

$$\mathbb{E} [\|\nabla f(w_T)\|_2^2] \leq \tilde{C}_1 \sqrt{\frac{d}{T_M}} + \tilde{C}_2 \frac{1}{T_M},$$

494 where T is a random termination number distributed according (4). The constants are defined as:

$$\begin{aligned} \tilde{C}_1 &= \frac{M}{(1 - a_m \beta_1) + (\beta_1 + a_m)} \left[\frac{a_m (1 - \beta_1)^2}{1 - \beta_2} + 2L \frac{1}{1 - \beta_2} + \Delta f + \frac{4L \beta_1^2 (1 + \beta_1^2)}{(1 - \beta_1)(1 - \beta_2)(1 - \gamma)} \right], \\ \tilde{C}_2 &= \frac{(a_m \beta_1^2 - 2a_m \beta_1 + \beta_1) M^2}{(1 - \beta_1)((1 - a_m \beta_1) + (\beta_1 + a_m))} \mathbb{E} [\|\hat{v}_0^{-1/2}\|], \end{aligned}$$

495 where $\Delta f = f(\bar{w}_1) - f(\bar{w}_{T_M+1})$ and $a_m = \min_{t=1, \dots, T} a_t$.

496 **Proof** Using H2 and the iterate \bar{w}_t we have:

$$\begin{aligned} f(\bar{w}_{t+1}) &\leq f(\bar{w}_t) + \nabla f(\bar{w}_t)^\top (\bar{w}_{t+1} - \bar{w}_t) + \frac{L}{2} \|\bar{w}_{t+1} - \bar{w}_t\|^2 \\ &\leq f(\bar{w}_t) + \underbrace{\nabla f(w_t)^\top (\bar{w}_{t+1} - \bar{w}_t)}_A \\ &\quad + \underbrace{(\nabla f(\bar{w}_t) - \nabla f(w_t))^\top (\bar{w}_{t+1} - \bar{w}_t)}_B + \frac{L}{2} \|\bar{w}_{t+1} - \bar{w}_t\|. \end{aligned} \quad (27)$$

497 **Term A.** Using Lemma 3, we have that:

$$\begin{aligned} \nabla f(w_t)^\top (\bar{w}_{t+1} - \bar{w}_t) &\leq \nabla f(w_t)^\top \left[\frac{\beta_1}{1 - \beta_1} \tilde{\theta}_{t-1} \left[\eta_{t-1} \hat{v}_{t-1}^{-1/2} - \eta_t \hat{v}_t^{-1/2} \right] - \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \right] \\ &\leq \frac{\beta_1}{1 - \beta_1} \|\nabla f(w_t)\| \|\eta_{t-1} \hat{v}_{t-1}^{-1/2} - \eta_t \hat{v}_t^{-1/2}\| \|\tilde{\theta}_{t-1}\| - \nabla f(w_t)^\top \eta_t \hat{v}_t^{-1/2} \tilde{g}_t, \end{aligned}$$

498 where the inequality is due to trivial inequality for positive diagonal matrix. Using Lemma 1 and
499 assumption H3 we obtain:

$$\nabla f(w_t)^\top (\bar{w}_{t+1} - \bar{w}_t) \leq \frac{\beta_1 (1 + \beta_1)}{1 - \beta_1} M^2 [\|\eta_{t-1} \hat{v}_{t-1}^{-1/2}\| - \|\eta_t \hat{v}_t^{-1/2}\|] - \nabla f(w_t)^\top \eta_t \hat{v}_t^{-1/2} \tilde{g}_t, \quad (28)$$

500 where we have used the fact that $\eta_t \hat{v}_t^{-1/2}$ is a diagonal matrix such that $\eta_{t-1} \hat{v}_{t-1}^{-1/2} \succcurlyeq \eta_t \hat{v}_t^{-1/2} \succcurlyeq 0$
501 (decreasing stepsize and max operator). Also note that:

$$\begin{aligned} -\nabla f(w_t)^\top \eta_t \hat{v}_t^{-1/2} \tilde{g}_t &= -\nabla f(w_t)^\top \eta_{t-1} \hat{v}_{t-1}^{-1/2} \tilde{g}_t - \nabla f(w_t)^\top \left[\eta_t \hat{v}_t^{-1/2} - \eta_{t-1} \hat{v}_{t-1}^{-1/2} \right] \tilde{g}_t \\ &\quad - \nabla f(w_t)^\top \eta_{t-1} \hat{v}_{t-1}^{-1/2} (\beta_1 g_{t-1} + m_{t+1}) \\ &\leq -\nabla f(w_t)^\top \eta_{t-1} \hat{v}_{t-1}^{-1/2} \tilde{g}_t + (1 - a_t \beta_1) M^2 [\|\eta_{t-1} \hat{v}_{t-1}^{-1/2}\| - \|\eta_t \hat{v}_t^{-1/2}\|] \\ &\quad - \nabla f(w_t)^\top \eta_t \hat{v}_t^{-1/2} (\beta_1 g_{t-1} + m_{t+1}), \end{aligned} \quad (29)$$

where we have used Lemma 1 on $\|g_t\|$ and where that $\tilde{g}_t = \bar{g}_t + \beta_1 g_{t-1} + m_{t+1} = g_t - \beta_1 m_t + \beta_1 g_{t-1} + m_{t+1}$. Plugging (29) into (28) yields:

$$\begin{aligned} & \nabla f(w_t)^\top (\bar{w}_{t+1} - \bar{w}_t) \\ & \leq -\nabla f(w_t)^\top \eta_{t-1} \hat{v}_{t-1}^{-1/2} \bar{g}_t + \frac{1}{1-\beta_1} (a_t \beta_1^2 - 2a_t \beta_1 + \beta_1) \mathbf{M}^2 [\|\eta_{t-1} \hat{v}_{t-1}^{-1/2}\| - \|\eta_t \hat{v}_t^{-1/2}\|] \\ & \quad - \nabla f(w_t)^\top \eta_t \hat{v}_t^{-1/2} (\beta_1 g_{t-1} + m_{t+1}). \end{aligned} \quad (30)$$

Term B. By Cauchy-Schwarz (CS) inequality we have:

$$(\nabla f(\bar{w}_t) - \nabla f(w_t))^\top (\bar{w}_{t+1} - \bar{w}_t) \leq \|\nabla f(\bar{w}_t) - \nabla f(w_t)\| \|\bar{w}_{t+1} - \bar{w}_t\|. \quad (31)$$

Using smoothness assumption H2:

$$\begin{aligned} \|\nabla f(\bar{w}_t) - \nabla f(w_t)\| & \leq L \|\bar{w}_t - w_t\| \\ & \leq L \frac{\beta_1}{1-\beta_1} \|w_t - \tilde{w}_{t-1}\|. \end{aligned} \quad (32)$$

By Lemma 3 we also have:

$$\begin{aligned} \bar{w}_{t+1} - \bar{w}_t & = \frac{\beta_1}{1-\beta_1} \tilde{\theta}_{t-1} \left[\eta_{t-1} \hat{v}_{t-1}^{-1/2} - \eta_t \hat{v}_t^{-1/2} \right] - \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \\ & = \frac{\beta_1}{1-\beta_1} \tilde{\theta}_{t-1} \eta_{t-1} \hat{v}_{t-1}^{-1/2} \left[I - (\eta_t \hat{v}_t^{-1/2})(\eta_{t-1} \hat{v}_{t-1}^{-1/2})^{-1} \right] - \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \\ & = \frac{\beta_1}{1-\beta_1} \left[I - (\eta_t \hat{v}_t^{-1/2})(\eta_{t-1} \hat{v}_{t-1}^{-1/2})^{-1} \right] (\tilde{w}_{t-1} - w_t) - \eta_t \hat{v}_t^{-1/2} \tilde{g}_t, \end{aligned} \quad (33)$$

where the last equality is due to $\tilde{\theta}_{t-1} \eta_{t-1} \hat{v}_{t-1}^{-1/2} = \tilde{w}_{t-1} - w_t$ by construction of $\tilde{\theta}_t$. Taking the norms on both sides, observing $\|I - (\eta_t \hat{v}_t^{-1/2})(\eta_{t-1} \hat{v}_{t-1}^{-1/2})^{-1}\| \leq 1$ due to the decreasing stepsize and the construction of \hat{v}_t and using CS inequality yield:

$$\|\bar{w}_{t+1} - \bar{w}_t\| \leq \frac{\beta_1}{1-\beta_1} \|\tilde{w}_{t-1} - w_t\| + \|\eta_t \hat{v}_t^{-1/2} \tilde{g}_t\|. \quad (34)$$

We recall Young's inequality with a constant $\delta \in (0, 1)$ as follows:

$$\langle X | Y \rangle \leq \frac{1}{\delta} \|X\|^2 + \delta \|Y\|^2.$$

Plugging (32) and (34) into (31) returns:

$$\begin{aligned} (\nabla f(\bar{w}_t) - \nabla f(w_t))^\top (\bar{w}_{t+1} - \bar{w}_t) & \leq L \frac{\beta_1}{1-\beta_1} \|\eta_t \hat{v}_t^{-1/2} \tilde{g}_t\| \|w_t - \tilde{w}_{t-1}\| \\ & \quad + L \left(\frac{\beta_1}{1-\beta_1} \right)^2 \|\tilde{w}_{t-1} - w_t\|^2. \end{aligned}$$

Applying Young's inequality with $\delta \rightarrow \frac{\beta_1}{1-\beta_1}$ on the product $\|\eta_t \hat{v}_t^{-1/2} \tilde{g}_t\| \|w_t - \tilde{w}_{t-1}\|$ yields:

$$(\nabla f(\bar{w}_t) - \nabla f(w_t))^\top (\bar{w}_{t+1} - \bar{w}_t) \leq L \|\eta_t \hat{v}_t^{-1/2} \tilde{g}_t\|^2 + 2L \left(\frac{\beta_1}{1-\beta_1} \right)^2 \|\tilde{w}_{t-1} - w_t\|^2. \quad (35)$$

The last term $\frac{L}{2} \|\bar{w}_{t+1} - \bar{w}_t\|^2$ can be upper bounded using (34):

$$\begin{aligned} \frac{L}{2} \|\bar{w}_{t+1} - \bar{w}_t\|^2 & \leq \frac{L}{2} \left[\frac{\beta_1}{1-\beta_1} \|\tilde{w}_{t-1} - w_t\| + \|\eta_t \hat{v}_t^{-1/2} \tilde{g}_t\| \right]^2 \\ & \leq L \|\eta_t \hat{v}_t^{-1/2} \tilde{g}_t\|^2 + 2L \left(\frac{\beta_1}{1-\beta_1} \right)^2 \|\tilde{w}_{t-1} - w_t\|^2. \end{aligned} \quad (36)$$

513 Plugging (30), (35) and (36) into (27) and taking the expectations on both sides give:

$$\begin{aligned} & \mathbb{E} \left[f(\bar{w}_{t+1}) + \frac{1}{1-\beta_1} \tilde{M}_t^2 \|\eta_t \hat{v}_t^{-1/2}\| - \left(f(\bar{w}_t) + \frac{1}{1-\beta_1} \tilde{M}_t^2 \|\eta_{t-1} \hat{v}_{t-1}^{-1/2}\| \right) \right] \\ & \leq \mathbb{E} \left[-\nabla f(w_t)^\top \eta_{t-1} \hat{v}_{t-1}^{-1/2} \tilde{g}_t - \nabla f(w_t)^\top \eta_t \hat{v}_t^{-1/2} (\beta_1 g_{t-1} + m_{t+1}) \right] \\ & + \mathbb{E} \left[2L \|\eta_t \hat{v}_t^{-1/2} \tilde{g}_t\|^2 + 4L \left(\frac{\beta_1}{1-\beta_1} \right)^2 \|\tilde{w}_{t-1} - w_t\|^2 \right], \end{aligned}$$

514 where $\tilde{M}_t^2 = (a_t \beta_1^2 + \beta_1) M^2$. Note that the expectation of \tilde{g}_t conditioned on the filtration \mathcal{F}_t reads
515 as follows

$$\mathbb{E} [\nabla f(w_t)^\top \tilde{g}_t] = \mathbb{E} [\nabla f(w_t)^\top (g_t - \beta_1 m_t)] = (1 - a_t \beta_1) \|\nabla f(w_t)\|^2. \quad (37)$$

516 Summing from $t = 1$ to $t = T$ leads to

$$\begin{aligned} & \frac{1}{M} \sum_{t=1}^{T_M} ((1 - a_t \beta_1) \eta_{t-1} + (\beta_1 + a_t) \eta_t) \|\nabla f(w_t)\|^2 \leq \\ & \mathbb{E} \left[f(\bar{w}_1) + \frac{1}{1-\beta_1} \tilde{M}_t^2 \|\eta_0 \hat{v}_0^{-1/2}\| - \left(f(\bar{w}_{T_M+1}) + \frac{1}{1-\beta_1} \tilde{M}_t^2 \|\eta_{T_M} \hat{v}_{T_M}^{-1/2}\| \right) \right] \\ & + 2L \sum_{t=1}^{T_M} \mathbb{E} [\|\eta_t \hat{v}_t^{-1/2} \tilde{g}_t\|^2] + 4L \left(\frac{\beta_1}{1-\beta_1} \right)^2 \sum_{t=1}^{T_M} \mathbb{E} [\|\tilde{w}_{t-1} - w_t\|^2] \quad (38) \\ & \leq \mathbb{E} \left[\Delta f + \frac{1}{1-\beta_1} \tilde{M}_t^2 \|\eta_0 \hat{v}_0^{-1/2}\| \right] + 2L \sum_{t=1}^{T_M} \mathbb{E} [\|\eta_t \hat{v}_t^{-1/2} \tilde{g}_t\|^2] \\ & + 4L \left(\frac{\beta_1}{1-\beta_1} \right)^2 \sum_{t=1}^{T_M} \mathbb{E} [\|\tilde{w}_{t-1} - w_t\|^2], \end{aligned}$$

517 where we denote $\Delta f := f(\bar{w}_1) - f(\bar{w}_{T_M+1})$. We note that by definition of \hat{v}_t , and a constant
518 learning rate η_t , we have

$$\begin{aligned} \|\tilde{w}_{t-1} - w_t\|^2 &= \|\eta_{t-1} \hat{v}_{t-1}^{-1/2} (\theta_{t-1} + h_t)\|^2 \\ &= \|\eta_{t-1} \hat{v}_{t-1}^{-1/2} (\theta_{t-1} + \beta_1 \theta_{t-2} + (1 - \beta_1) m_t)\|^2 \\ &\leq \|\eta_{t-1} \hat{v}_{t-1}^{-1/2} \theta_{t-1}\|^2 + \|\eta_{t-2} \hat{v}_{t-2}^{-1/2} \beta_1 \theta_{t-2}\|^2 + (1 - \beta_1)^2 \|\eta_{t-1} \hat{v}_{t-1}^{-1/2} m_t\|^2. \end{aligned}$$

519 Using Lemma 4 we have

$$\begin{aligned} & \sum_{t=1}^{T_M} \mathbb{E} [\|\tilde{w}_{t-1} - w_t\|^2] \\ & \leq (1 + \beta_1^2) \frac{\eta^2 d T_M (1 - \beta_1)}{(1 - \beta_2)(1 - \gamma)} + (1 - \beta_1)^2 \sum_{t=1}^{T_M} \mathbb{E} [\|\eta_{t-1} \hat{v}_{t-1}^{-1/2} m_t\|]. \end{aligned}$$

520 Assume $a_m = \min_{1, \dots, T_M} a_t$ and denote $\tilde{M}_m^2 = (a_m \beta_1^2 + \beta_1) M^2$. Setting a constant learning rate
 521 $\eta_t = \eta$ and plugging in (38) yields:

$$\begin{aligned} \mathbb{E}[\|\nabla f(w_T)\|^2] &= \frac{1}{\sum_{j=1}^{T_M} \eta_j} \sum_{t=1}^{T_M} \eta_t \|\nabla f(w_t)\|^2 = \frac{\sum_{t=1}^{T_M} \|\nabla f(w_t)\|^2}{T_M} \\ &\leq \frac{M}{T_M \eta ((1 - a_m \beta_1) + (\beta_1 + a_m))} \mathbb{E} \left[\Delta f + \frac{1}{1 - \beta_1} \tilde{M}_m^2 \|\eta_0 \hat{v}_0^{-1/2}\| \right] \\ &\quad + \frac{4L \left(\frac{\beta_1}{1 - \beta_1} \right)^2 M}{T_M \eta ((1 - a_m \beta_1) + (\beta_1 + a_m))} (1 + \beta_1^2) \frac{\eta^2 d T_M (1 - \beta_1)}{(1 - \beta_2)(1 - \gamma)} \\ &\quad + \frac{M}{T_M \eta ((1 - a_m \beta_1) + (\beta_1 + a_m))} (1 - \beta_1)^2 \sum_{t=1}^{T_M} \mathbb{E}[\|\eta_{t-1} \hat{v}_{t-1}^{-1/2} m_t\|] \\ &\quad + \frac{2LM}{T_M \eta ((1 - a_m \beta_1) + (\beta_1 + a_m))} \sum_{t=1}^{T_M} \mathbb{E}[\|\eta_t \hat{v}_t^{-1/2} \tilde{g}_t\|^2], \end{aligned}$$

522 where T is a random termination number distributed according (4) and T_M is the maximum number
 523 of iteration. Setting the stepsize to $\eta = \frac{1}{\sqrt{dT_M}}$ yields :

$$\mathbb{E}[\|\nabla f(w_T)\|^2] \leq C_{1,m} \sqrt{\frac{d}{T_M}} + C_{2,m} \frac{1}{T_M} + \frac{\eta}{T_M} D_{1,m} \mathbb{E}[\|\hat{v}_{t-1}^{-1/2} m_t\|] + \frac{\eta}{T_M} D_{2,m} \mathbb{E}[\|\hat{v}_{t-1}^{-1/2} \tilde{g}_t\|],$$

524 where

$$\begin{aligned} C_{1,m} &= \frac{M}{(1 - a_m \beta_1) + (\beta_1 + a_m)} \Delta f + \frac{4L \left(\frac{\beta_1}{1 - \beta_1} \right)^2 M}{(1 - a_m \beta_1) + (\beta_1 + a_m)} \frac{(1 + \beta_1^2)(1 - \beta_1)}{(1 - \beta_2)(1 - \gamma)}, \\ C_{2,m} &= \frac{M}{(1 - \beta_1)((1 - a_m \beta_1) + (\beta_1 + a_m))} (a_m \beta_1^2 + \beta_1) M^2 \mathbb{E}[\|\hat{v}_0^{-1/2}\|]. \end{aligned}$$

525 **Simple case as in [Zhou et al., 2018]:** if $\beta_1 = 0$ then $\tilde{g}_t = g_t + m_{t+1}$ and $g_t = \theta_t$. Also using
 526 Lemma 4 we have that:

$$\sum_{t=1}^{T_M} \eta_t^2 \mathbb{E} \left[\left\| \hat{v}_t^{-1/2} g_t \right\|_2^2 \right] \leq \frac{\eta^2 d T_M}{(1 - \beta_2)};$$

527 which leads to the final bound:

$$\mathbb{E}[\|\nabla f(w_T)\|^2] \leq \sqrt{\frac{d}{T_M}} \tilde{C}_{1,m} + \frac{1}{T_M} \tilde{C}_{2,m},$$

528 where

$$\begin{aligned} \tilde{C}_{1,m} &= C_{1,m} + \frac{M}{(1 - a_m \beta_1) + (\beta_1 + a_m)} \left[\frac{a_m (1 - \beta_1)^2}{1 - \beta_2} + 2L \frac{1}{1 - \beta_2} \right], \\ \tilde{C}_{2,m} &= C_{2,m} = \frac{M}{(1 - \beta_1)((1 - a_m \beta_1) + (\beta_1 + a_m))} \tilde{M}_m^2 \mathbb{E}[\|\hat{v}_0^{-1/2}\|]. \end{aligned}$$

529

□

530 E Proof of Lemma 2 (Boundedness of the iterates H1)

531 **Lemma.** Given the multilayer model (5), assume the boundedness of the input data and of the loss
 532 function, i.e., for any $\xi \in \mathbb{R}^p$ and $y \in \mathbb{R}$ there is a constant $T > 0$ such that:

$$\|\xi\| \leq 1 \quad \text{a.s.} \quad \text{and} \quad |\mathcal{L}'(\cdot, y)| \leq T, \quad (39)$$

where $\mathcal{L}'(\cdot, y)$ denotes its derivative w.r.t. the parameter. Then for each layer $\ell \in [1, L]$, there exist
 a constant $A_{(\ell)}$ such that:

$$\|w^{(\ell)}\| \leq A_{(\ell)}.$$

Proof For any index $\ell \in [1, L]$ we denote the output of layer ℓ by

$$h^{(\ell)}(w, \xi) = \sigma \left(w^{(\ell)} \sigma \left(w^{(\ell-1)} \dots \sigma \left(w^{(1)} \xi \right) \right) \right) .$$

533 Given the sigmoid assumption we have $\|h^{(\ell)}(w, \xi)\| \leq 1$ for any $\ell \in [1, L]$ and any $(w, \xi) \in$
 534 $\mathbb{R}^d \times \mathbb{R}^p$. We also recall that $\mathcal{L}(\cdot, y)$ is the loss function, which can be Huber loss or cross entropy.

535 Observe that at the last layer L :

$$\begin{aligned} \|\nabla_{w^{(L)}} \mathcal{L}(\text{MLN}(w, \xi), y)\| &= \|\mathcal{L}'(\text{MLN}(w, \xi), y) \nabla_{w^{(L)}} \text{MLN}(w, \xi)\| \\ &= \|\mathcal{L}'(\text{MLN}(w, \xi), y) \sigma'(w^{(L)} h^{(L-1)}(w, \xi)) h^{(L-1)}(w, \xi)\| \\ &\leq \frac{T}{4} , \end{aligned} \quad (40)$$

536 where the last equality is due to mild assumptions (39) and to the fact that the norm of the derivative
 537 of the sigmoid function is upperbounded by $1/4$.

538 From Algorithm 2, and with $\beta_1 = 0$ for the sake of notation, we have for iteration index $t > 0$:

$$\begin{aligned} \|w_t - \tilde{w}_{t-1}\| &= \|\eta_t \hat{v}_t^{-1/2} (\theta_t + h_{t+1})\| = \|\eta_t \hat{v}_t^{-1/2} (g_t + m_{t+1})\| \\ &\leq \hat{\eta} \|\hat{v}_t^{-1/2} g_t\| + \hat{\eta} a \|\hat{v}_t^{-1/2} g_{t+1}\| , \end{aligned}$$

where $\hat{\eta} = \max_{t>0} \eta_t$. For any dimension $p \in [1, d]$, using assumption H3, we note that

$$\sqrt{\hat{v}_{t,p}} \geq \sqrt{1 - \beta_2} g_{t,p} \quad \text{and} \quad m_{t+1} \leq a \|g_{t+1}\| .$$

539 Thus:

$$\|w_t - \tilde{w}_{t-1}\| \leq \hat{\eta} \left(\|\hat{v}_t^{-1/2} g_t\| + a \|\hat{v}_t^{-1/2} g_{t+1}\| \right) \leq \hat{\eta} \frac{a+1}{\sqrt{1-\beta_2}} .$$

540 In short there exist a constant B such that $\|w_t - \tilde{w}_{t-1}\| \leq B$.

Proof by induction: As in [Défossez et al., 2020], we will prove the containment of the weights by induction. Suppose an iteration index T and a coordinate i of the last layer L such that $w_{T,i}^{(L)} \geq \frac{T}{4\lambda} + B$. Using (40), we have

$$\nabla_i f(w_t^{(L)}, \xi) \geq -\frac{T}{4} + \lambda \frac{T}{\lambda 4} \geq 0 ,$$

541 where $f(w, \xi) = \mathcal{L}(\text{MLN}(w, \xi), y) + \frac{\lambda}{2} \|w\|^2$ and is the loss of our MLN. This last equation yields
 542 $\theta_{T,i}^{(L)} \geq 0$ (given the algorithm and $\beta_1 = 0$) and using the fact that $\|w_t - \tilde{w}_{t-1}\| \leq B$ we have

$$0 \leq w_{T-1,i}^{(L)} - B \leq w_{T,i}^{(L)} \leq w_{T-1,i}^{(L)} , \quad (41)$$

which means that $|w_{T,i}^{(L)}| \leq w_{T-1,i}^{(L)}$. So if the first assumption of that induction reasoning holds, i.e., $w_{T-1,i}^{(L)} \geq \frac{T}{4\lambda} + B$, then the next iterates $w_{T,i}^{(L)}$ decreases, see (41) and go below $\frac{T}{4\lambda} + B$. This yields that for any iteration index $t > 0$ we have

$$w_{T,i}^{(L)} \leq \frac{T}{4\lambda} + 2B ,$$

since B is the biggest jump an iterate can do since $\|w_t - \tilde{w}_{t-1}\| \leq B$. Likewise we can end up showing that

$$|w_{T,i}^{(L)}| \leq \frac{T}{4\lambda} + 2B ,$$

543 meaning that the weights of the last layer at any iteration is bounded in some matrix norm.

544 Now that we have shown this boundedness property for the last layer L , we will do the same for the
 545 previous layers and conclude the verification of assumption H1 by induction.

546 For any layer $\ell \in [1, L - 1]$, we have:

$$\nabla_{w^{(\ell)}} \mathcal{L}(\text{MLN}(w, \xi), y) = \mathcal{L}'(\text{MLN}(w, \xi), y) \left(\prod_{j=1}^{\ell+1} \sigma' \left(w^{(j)} h^{(j-1)}(w, \xi) \right) \right) h^{(\ell-1)}(w, \xi). \quad (42)$$

This last quantity is bounded as long as we can prove that for any layer ℓ the weights $w^{(\ell)}$ are bounded in some matrix norm as $\|w^{(\ell)}\|_F \leq F_\ell$ with the Frobenius norm. Suppose we have shown $\|w^{(r)}\|_F \leq F_r$ for any layer $r > \ell$. Then having this gradient (42) bounded we can use the same lines of proof for the last layer L and show that the norm of the weights at the selected layer ℓ satisfy

$$\|w^{(\ell)}\| \leq \frac{T \prod_{t \geq \ell} F_t}{4^{L-\ell+1}} + 2B.$$

547 Showing that the weights of the previous layers $\ell \in [1, L - 1]$ as well as for the last layer L of our
 548 fully connected feed forward neural network are bounded at each iteration, leads by induction, to
 549 the boundedness (at each iteration) assumption we want to check, thus proving Lemma 2. \square

550 F Additional Remarks and Runs on the Gradient Prediction Process

Two illustrative examples. We provide two toy examples to demonstrate how OPT-AMSGRAD works with the chosen extrapolation method. First, consider minimizing a quadratic function $H(w) := \frac{b}{2}w^2$ with vanilla gradient descent method $w_{t+1} = w_t - \eta_t \nabla H(w_t)$. The gradient $g_t := \nabla H(w_t)$ can be recursively expressed as $g_{t+1} = bw_{t+1} = b(w_t - \eta_t g_t) = g_t - b\eta_t g_t$. Thus, the update can be written in the form of

$$g_t = Ag_{t-1} + \mathcal{O}(\|g_{t-1}\|_2^2)u_{t-1},$$

551 where $A = (1 - b\eta)$ and $u_{t-1} = 0$ by setting $\eta_t = \eta$ (constant step size). Therefore, the extrapolation
 552 method should predict well. Specifically, consider optimizing $H(w) := w^2/2$ by the following
 553 three algorithms with the same step size. One is Gradient Descent (GD): $w_{t+1} = w_t - \eta_t g_t$, while
 554 the other two are OPT-AMSGRAD with $\beta_1 = 0$ and the second moment term \hat{v}_t being dropped:
 555 $w_{t+\frac{1}{2}} = \Pi_{\Theta}[w_{t-\frac{1}{2}} - \eta_t g_t]$, $w_{t+1} = \Pi_{\Theta}[w_{t+\frac{1}{2}} - \eta_{t+1} m_{t+1}]$. We denote the algorithm that sets
 556 $m_{t+1} = g_t$ as OPT-1, and denote the algorithm that uses the extrapolation method to get m_{t+1} as
 557 OPT-EXTRA. We let $\eta_t = 0.1$ and the initial point $w_0 = 5$ for all three methods. The simulation
 558 results are on Figure 6 (a) and (b). Sub-figure (a) plots update w_t over iteration, where the updates
 559 should go towards the optimal point 0. Sub-figure (b) displays a scaled and clipped version of m_t ,
 560 defined as $w_t - w_{t-1/2}$, which can be viewed as $-\eta_t m_t$ if the projection (if exists) is lifted. Sub-
 561 figure (a) shows that OPT-EXTRA converges faster than the other methods. Furthermore, sub-figure
 562 (b) shows that the prediction by the extrapolation method is better than the prediction by simply
 563 using the previous gradient. The sub-figure shows that $-m_t$ from both methods points to 0 for each
 564 iteration and the magnitude is larger for the one produced by the extrapolation method after iteration
 565 2.²

566 Now let us consider another problem: an online learning problem proposed in [Reddi et al., 2018]
 567 ³. Assume the learner's decision space is $\Theta = [-1, 1]$, and the loss function is $\ell_t(w) = 3w$ if
 568 $t \bmod 3 = 1$, and $\ell_t(w) = -w$ otherwise. The optimal point to minimize the cumulative loss is
 569 $w^* = -1$. We let $\eta_t = 0.1/\sqrt{t}$ and the initial point $w_0 = 1$ for all three methods. The parameter λ
 570 of the extrapolation method is set to $\lambda = 10^{-3} > 0$. The results are reported Figure 6 (c) and (d).
 571 Sub-figure (c) shows that OPT-EXTRA converges faster than the other methods while OPT-1 is not
 572 performing better than GD. The reason is that the gradient changes from -1 to 3 at $t \bmod 3 = 1$ and
 573 it changes from 3 to -1 at $t \bmod 3 = 2$. Consequently, using the current gradient as the guess for the
 574 next is empirically not a good choice, since the next gradient is in the opposite direction of the current
 575 one, according to our experiments. Sub-figure (d) shows that $-m_t$, obtained with the extrapolation
 576 method, always points to $w^* = -1$, while the one obtained by using the previous negative direction
 577 points to the opposite direction in two thirds of rounds. It shows that the extrapolation method is
 578 much less affected by the gradient oscillation and always makes the prediction in the right direction,
 579 which suggests that the method can capture the aggregate effect.

²The extrapolation needs at least two gradients for prediction. Thus, in the first two iterations, $m_t = 0$.

³[Reddi et al., 2018] uses this example to show that ADAM [Kingma and Ba, 2015] fails to converge.

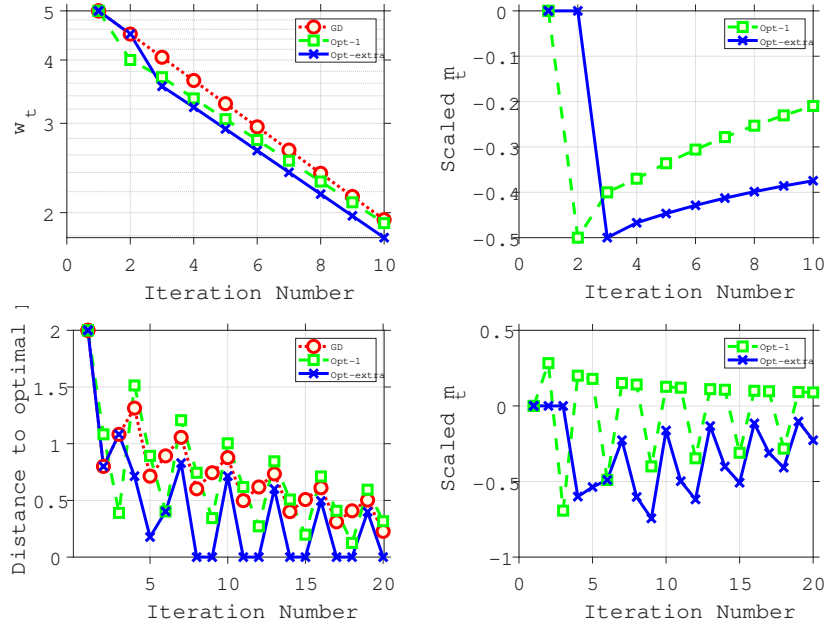


Figure 6: (a): The iterate w_t ; the closer to the optimal point 0 the better. (b): A scaled and clipped version of m_t : $w_t - w_{t-1/2}$, which measures how the prediction of m_t drives the update towards the optimal point. In this scenario, the more negative the better. (c): Distance to the optimal point -1 . The smaller the better. (d): A scaled and clipped version of m_t : $w_t - w_{t-1/2}$, which measures how the prediction of m_t drives the update towards the optimal point. In this scenario, the more negative the better.