

---

# OPT-AMSGrad: An Optimistic Acceleration of AMSGrad for Nonconvex Optimization

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

1 In this paper, we propose a new variant of AMSGrad [31], a popular adaptive gra-  
2 dent based optimization algorithm widely used in training deep neural networks.  
3 Our algorithm adds prior knowledge about the sequence of consecutive mini-batch  
4 gradients and leverages its underlying structure making the gradients sequentially  
5 predictable. By exploiting the predictability and ideas from Optimistic Online  
6 Learning, the proposed algorithm can accelerate the convergence and increase  
7 sample efficiency. After establishing a tighter upper bound under some convexity  
8 conditions on the regret, we offer a complimentary view of our algorithm which  
9 generalizes the offline and stochastic version of nonconvex optimization. In the  
10 nonconvex case, we establish a  $\mathcal{O}\left(\sqrt{d/T} + d/T\right)$  non-asymptotic bound inde-  
11 pendently of the initialization of the method. We illustrate the practical speedup  
12 on several deep learning models through numerical experiments.

## 13 1 Introduction

14 Deep learning models have been successful in several applications, from robotics (e.g. [21]), com-  
15 puter vision (e.g. [18, 15]), reinforcement learning (e.g. [25]), to natural language processing (e.g.  
16 [16]). With the sheer size of modern data sets and the dimension of neural networks, speeding up  
17 training is of utmost importance. To do so, several algorithms have been proposed in recent years,  
18 such as AMSGRAD [31], ADAM [19], RMSPROP [35], ADADELTA [41], and NADAM [10].

19 All the prevalent algorithms for training deep networks mentioned above combine two ideas: the  
20 idea of adaptivity from ADAGRAD [11, 23] and the idea of momentum from NESTEROV’S METHOD  
21 [27] or HEAVY BALL method [28]. ADAGRAD is an online learning algorithm that works well  
22 compared to the standard online gradient descent when the gradient is sparse. Its update has a  
23 notable feature: it leverages an anisotropic learning rate depending on the magnitude of gradient in  
24 each dimension which helps in exploiting the geometry of data. On the other hand, NESTEROV’S  
25 METHOD or HEAVY BALL Method [28] is an accelerated optimization algorithm which update not  
26 only depends on the current iterate and current gradient but also depends on the past gradients (i.e.  
27 momentum). State-of-the-art algorithms like AMSGRAD [31] and ADAM [19] leverage these ideas  
28 to accelerate the training of nonconvex objective functions such as deep neural networks losses.

29 In this paper, we propose an algorithm that goes further than the hybrid of the adaptivity and mo-  
30 mentum approach. Our algorithm is inspired by OPTIMISTIC ONLINE LEARNING [7, 29, 34, 1, 24],  
31 which assumes that, in each round of online learning, a *predictable process* of the gradient of the  
32 loss function is available. Then an action is played exploiting these predictors. By capitalizing on  
33 this (possibly) arbitrary process, algorithms in OPTIMISTIC ONLINE LEARNING enjoy smaller re-  
34 gret than the ones gradient predictions. We combine the OPTIMISTIC ONLINE LEARNING idea with  
35 the adaptivity and the momentum ideas to design a new algorithm — OPT-AMSGRAD.

A single work along that direction stands out. [8] develops OPTIMISTIC-ADAM leveraging optimistic online mirror descent [30]. Yet, OPTIMISTIC-ADAM is specifically designed to optimize two-player games, e.g. GANs [15] which is in particular a two-player zero-sum game. There have been some related works in OPTIMISTIC ONLINE LEARNING like [7, 30, 34] showing that if both players use an OPTIMISTIC type of update, then accelerating the convergence to the equilibrium of the game is possible. Authors in [8] build on these related works and show that OPTIMISTIC-MIRROR-DESCENT can avoid the cycle behavior in a bilinear zero-sum game, which accelerates the convergence. In contrast, in this paper, the proposed algorithm is designed to accelerate nonconvex optimization (e.g. empirical risk minimization). To the best of our knowledge, this is the first work exploring towards this direction and bridging the unfilled *theoretical* gap at the crossroads of online learning and stochastic optimization. The contributions of this paper are as follows:

- We derive an optimistic variant of AMSGRAD borrowing techniques from online learning procedures. Our method relies on (I) the addition of *prior knowledge* in the sequence of the model parameter estimations alleviating a predictable process able to provide guesses of gradients of the loss functions through the iterations and (II) the construction of a *double update* algorithm done sequentially. We interpret this two-projection step as the learning of the global parameter and of an underlying scheme which makes the gradients sequentially predictable.
- We focus on the *theoretical* justifications of our method by establishing novel *non-asymptotic* and *global* convergence rates in both convex and nonconvex cases. Based on *convex regret minimization* and *nonconvex stochastic optimization* views, we prove, respectively, that our algorithm suffers regret of  $\mathcal{O}\left(\sqrt{\sum_{t=1}^T \|g_t - m_t\|_{\psi_{t-1}}^2}\right)$  and achieves a rate of convergence  $\mathcal{O}\left(\sqrt{d/T} + d/T\right)$ .

The proposed algorithm not only adapts to the informative dimensions, exhibits momentum, but also exploits a good guess of the next gradient to facilitate acceleration. Besides the global analysis of OPT-AMSGRAD, we conduct experiments and show that the proposed algorithm not only accelerates the training procedure, but also leads to better empirical generalization performance.

Section 2 is devoted to introductory notions on online learning for regret minimization and adaptive learning methods for nonconvex stochastic optimization. We introduce in Section 3 our new algorithm, namely OPT-AMSGRAD and provide a comprehensive global analysis in both *convex/online* and *nonconvex/offline* settings in Section 4. We illustrate the benefits of our method on several finite-sum nonconvex optimization problems in Section 5. The supplementary material of this paper is devoted to the proofs of our theoretical results.

**Notations:** We follow the notations in related adaptive optimization papers [19, 31]. For any vector  $u, v \in \mathbb{R}^d$ ,  $u/v$  represents element-wise division,  $u^2$  represents element-wise square,  $\sqrt{u}$  represents element-wise square-root. We denote  $g_{1:T}[i]$  as the sum of the  $i_{th}$  element of  $g_1, g_2, \dots, g_T \in \mathbb{R}^d$ .

## 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 of the learner is to minimize the regret,

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

which is the cumulative loss of the learner minus the cumulative loss of some benchmark  $w^* \in \Theta$ . The idea of OPTIMISTIC ONLINE LEARNING (e.g. [7, 29, 34, 1]) is as follows. In each round  $t$ , the learner exploits a guess  $m_t(\cdot)$  of the gradient  $\nabla \ell_t(\cdot)$  of the loss function to choose an action  $w_t$ <sup>1</sup>. Consider the FOLLOW-THE-REGULARIZED-LEADER (FTRL, [17]) online learning algorithm which update reads

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

<sup>1</sup>Imagine 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  $\eta$  is a parameter,  $\mathbf{R}(\cdot)$  is a 1-strongly convex function with respect to a given norm on the constraint set  $\Theta$ , 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, noted OPTIMISTIC-FTRL and developed in [34] reads

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

where  $\{m_t\}_{t>0}$  is a predictable process incorporating (possibly arbitrarily) knowledge about the sequence of gradients  $\{g_t := \nabla \ell_t(w_t)\}_{t>0}$ . Under the assumption that loss functions are convex, 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 [29]. Yet, when the predictions are well designed, the regret will be lower. We will have a similar argument when we compare OPT-AMSGRAD and AMSGRAD.

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 5 a simple, yet effective, predictable process  $\{m_t\}_{t>0}$  leading to empirical acceleration.

**Adaptive optimization methods.** Adaptive optimization has been popular in various deep learning applications due to their superior empirical performance. ADAM [19], a popular adaptive algorithm, combines momentum [28] and anisotropic learning rate of ADAGRAD [11]. More specifically, the learning rate of ADAGRAD at time  $t$  for dimension  $j$  is proportional to the inverse of  $\sqrt{\sum_{s=1}^t g_s[j]^2}$ , where  $g_s[j]$  is the  $j$ -th element of the gradient vector  $g_s$  at time  $s$ . This adaptive learning rate helps accelerating the convergence when the gradient vector is sparse [11] but, when applying ADAGRAD to train deep neural networks, it is observed that the learning rate might decay too fast [19]. Therefore, [19] proposes ADAM that uses a moving average of gradients divided by the square root of the second moment of the moving average (element-wise multiplication), for updating the model parameter  $w$ . A variant, called AMSGRAD and detailed in Algorithm 1, has been developed in [31] to fix ADAM failures. The difference between ADAM and AMSGRAD lies in line 7 of Algorithm 1. AMSGRAD [31] adds the max operation to guarantee a non-increasing learning rate  $\eta_t/\sqrt{\hat{v}_t}$ , which helps for the convergence (i.e. average regret  $\mathcal{R}_T/T \rightarrow 0$ ).

---

#### Algorithm 1 AMSGRAD [31]

---

```

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
```

---

### 3 OPT-AMSGRAD Algorithm

We formulate in this section the proposed optimistic acceleration of AMSGrad, noted 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 it maintains an exponential moving average of  $\theta_t \in \mathbb{R}^d$  (line 5) and  $v_t \in \mathbb{R}^d$  (line 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) and 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 update  $w_{t+1} = \tilde{w}_t - \eta_t(\theta_t + h_{t+1})/\sqrt{\hat{v}_t}$ .

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 [29]. It leverages the auxiliary variable (hidden model) to update and commit  $w_{t+1}$ , which exploits the guess  $m_{t+1}$ , see Figure 1. In the following analysis, we show that the 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 [7]. The gradient prediction process plays an important role as discussed in Section 5.

---

**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  $v_0 = \epsilon \mathbf{1} \in \mathbb{R}^d$ .  
 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:    $\tilde{w}_{t+1} = \tilde{w}_t - \eta_t \frac{\theta_t}{\sqrt{\hat{v}_t}}$ .  
 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**

---

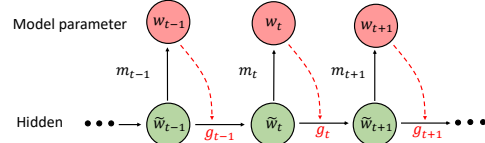


Figure 1: OPT-AMSGRAD UNDERLYING STRUCTURE.

130 The proposed OPT-AMSGRAD inherits three properties:

- 131 • Adaptive learning rate of each dimension as ADAGRAD [11]. (line 6, line 8 and line 9)
- 132 • Exponential moving average of the past gradients as NESTEROV'S METHOD [27] and the
- 133 HEAVY-BALL method [28]. (line 5)
- 134 • Optimistic update that exploits *prior knowledge* of the next gradient vector as in optimistic
- 135 online learning algorithms [7, 29, 34]. (line 9)

136 The first property helps for acceleration when the gradient has a sparse structure. The second one is  
 137 from the long-established idea of momentum which can also help for acceleration. The last one can  
 138 lead to an acceleration when the prediction of the next gradient is good as mentioned above when  
 139 introducing the regret bound for the OPTIMISTIC-FTRL algorithm. This property will be elaborated  
 140 whilst establishing the theoretical analysis of OPT-AMSGRAD.

## 141 4 Global Convergence of OPT-AMSGRAD

142 For conciseness, we place all the proofs of the following results in the supplementary material.

143 **Notations.** We denote the Mahalanobis norm  $\|\cdot\|_H := \sqrt{\langle \cdot, H \cdot \rangle}$  for some positive semidefinite  
 144 (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  
 145  $\text{diag}\{\hat{v}_t\}$  represents the diagonal matrix which  $i_{th}$  diagonal element is  $\hat{v}_t[i]$  defined in Algorithm 2.  
 146 We define its corresponding Mahalanobis norm  $\|\cdot\|_{\psi_t} := \sqrt{\langle \cdot, \text{diag}\{\hat{v}_t\}^{1/2} \cdot \rangle}$ , where we abuse  
 147 the notation  $\psi_t$  to represent the PSD matrix  $H_t^{1/2} := \text{diag}\{\hat{v}_t\}^{1/2}$ . Note that  $\psi_t(\cdot)$  is 1-strongly  
 148 convex with respect to the norm  $\|\cdot\|_{\psi_t}$ . Namely,  $\psi_t(\cdot)$  satisfies  $\psi_t(u) \geq \psi_t(v) + \langle \psi_t(v), u -$   
 149  $v \rangle + \frac{1}{2} \|u - v\|_{\psi_t}^2$  for any point  $(u, v) \in \Theta^2$ . A consequence of 1-strongly convexity of  $\psi_t(\cdot)$  is  
 150 that  $B_{\psi_t}(u, v) \geq \frac{1}{2} \|u - v\|_{\psi_t}^2$ , where the Bregman divergence  $B_{\psi_t}(u, v)$  is defined as  $B_{\psi_t}(u, v) :=$   
 151  $\psi_t(u) - \psi_t(v) - \langle \psi_t(v), u - v \rangle$  with  $\psi_t(\cdot)$  as the distance generating function. We also define the  
 152 corresponding dual norm  $\|\cdot\|_{\psi_t^*} := \sqrt{\langle \cdot, \text{diag}\{\hat{v}_t\}^{-1/2} \cdot \rangle}$ .

### 153 4.1 Convex Regret Analysis

154 In this section, we assume that the loss functions  $\{\ell_t\}_{t>0}$  are convex. We also assume that  $\Theta$  has  
 155 bounded diameter  $D_\infty$ , which is a standard assumption in previous works [31, 19] on adaptive  
 156 methods. It is necessary in regret analysis since if the boundedness assumption is lifted, one might  
 157 construct a scenario such that the benchmark is  $w^* = \infty$  and the learner's regret is infinite.

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

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

160 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  $D_\infty^2$  is the diameter of  
 161 the bounded set  $\Theta$ . The result holds for any benchmark  $w^* \in \Theta$  and any step size sequence  $\{\eta_t\}_{t>0}$ .

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

$$\mathcal{R}_T \leq \frac{B_{\psi_1}(w^*, \tilde{w}_1)}{\eta_1} + \frac{\eta\sqrt{1+\log T}}{\sqrt{1-\beta_2}} \sum_{i=1}^d \|(g-m)_{1:T}[i]\|_2 + \frac{D_\infty^2}{\eta_{\min}} \sum_{i=1}^d \left[ (1-\beta_2) \sum_{s=1}^T \beta_2^{T-s} g_s[i]^2 \right]^{1/2},$$

164 where  $g_t := \nabla \ell_t(w_t)$  and  $\eta_{\min} := \min_t \eta_t$ .

165 We can compare the bound of Corollary 1 with that of AMSGRAD [31] with  $\eta_t = \eta/\sqrt{t}$ :

$$\mathcal{R}_T \leq \frac{\eta\sqrt{1+\log T}}{\sqrt{1-\beta_2}} \sum_{i=1}^d \|g_{1:T}[i]\|_2 + \frac{\sqrt{T}}{2\eta} D_\infty^2 \sum_{i=1}^d \hat{v}_T[i]^2. \quad (3)$$

166 For convex regret minimization, the results above yields that the learner suffers regret of  
 167  $\mathcal{O}(\sqrt{\sum_{t=1}^T \|g_t - m_t\|_{\psi_t^*}^2})$  with an access to an arbitrary predictable process  $\{m_t\}_{t>0}$  of the mini-  
 168 batch gradients. The better the predictors, the lower the regret, see the second term in Corollary 1  
 169 compared to the first term in (3). The construction of the predictable process  $\{m_t\}_{t>0}$  is thus of  
 170 utmost importance for achieving optimal acceleration and can be learned through the iterations. We  
 171 will not deal with the latter in this paper for the sake of space and clarity. Though, for implemen-  
 172 tation purposes, we derive a simple, yet effective, gradient prediction algorithm, see Algorithm 3 in  
 173 Section 5, embedded in our OPT-AMSGRAD algorithm.

## 174 4.2 Nonconvex Analysis (Finite-Time Upper Bound)

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

$$\min_{w \in \Theta} f(w) := \mathbb{E}[f(w, \xi)] = n^{-1} \sum_{i=1}^n \mathbb{E}[f(w, \xi_i)], \quad (4)$$

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

$$P(T = \ell) = \frac{\eta_\ell}{\sum_{j=0}^{T_M-1} \eta_j}, \quad (5)$$

180 where  $T_M$  is the maximum number of iteration. The random termination number (5) is inspired by  
 181 [14] and is widely used for nonconvex optimization. Assume the following:

182 **H1.** For any  $t > 0$ , the estimated weight  $w_t$  stays within a  $\ell_\infty$ -ball. There exists a constant  $W > 0$   
 183 such that  $\|w_t\| \leq W$  almost surely.

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

186 We assume that the optimistic guess  $m_t$  at iteration  $t$  and the true gradient  $g_t$  are correlated:

187 **H3.** There exists a constant  $a \in \mathbb{R}^*$  such that for any  $t > 0$ ,  $\langle m_t | g_t \rangle \leq a \|g_t\|^2$ .

188 Classically in nonconvex optimization [14] we make an assumption on the magnitude of the gradient:

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

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

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

194 Then, following [39] and their study of the SGD with Momentum we denote for any  $t > 0$ :

$$\bar{w}_t = w_t + \frac{\beta_1}{1-\beta_1} (w_t - \tilde{w}_{t-1}) = \frac{1}{1-\beta_1} w_t - \frac{\beta_1}{1-\beta_1} \tilde{w}_{t-1}, \quad (6)$$

195 **Lemma 2.** Assume a strictly positive and non increasing sequence of stepsizes  $\{\eta_t\}_{t>0}$ ,  $\beta_1 < \beta_2 \in$   
 196  $[0, 1)$ , then the following holds:

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

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

198 **Lemma 3.** Assume H4, a strictly positive and a sequence of constant stepsizes  $\{\eta_t\}_{t>0}$ ,  $\beta_1 < \beta_2 \in$   
 199  $[0, 1)$ , denote  $\gamma = \beta_1/\beta_2$ , then,  $\sum_{k=1}^{T_M} \eta_t^2 \mathbb{E}[\|\hat{v}_t^{-1/2} \theta_t\|_2^2] \leq \eta^2 d T_M (1 - \beta_1) / [(1 - \beta_2)(1 - \gamma)]$ .

200 We now formulate the main result of our paper yielding a finite-time upper bound of the subopti-  
 201 mality condition  $\mathbb{E}[\|\nabla f(w_T)\|^2]$  (as the convergence criterion of interest, see [14]):

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

$$\mathbb{E}[\|\nabla f(w_T)\|^2] \leq \tilde{C}_1 \sqrt{\frac{d}{T_M}} + \tilde{C}_2 \frac{1}{T_M} , \quad (7)$$

204 where  $T$  is a random termination number distributed according (5). The constants are defined as:

$$\begin{aligned} \tilde{C}_1 &= C_1 + \frac{M}{(1 - a\beta_1) + (\beta_1 + a)} \left[ \frac{a(1 - \beta_1)^2}{1 - \beta_2} + 2L \frac{1}{1 - \beta_2} + \Delta f + \frac{4L\beta_1^2(1 + \beta_1^2)}{(1 - \beta_1)(1 - \beta_2)(1 - \gamma)} \right] \\ \tilde{C}_2 &= \frac{(a\beta_1^2 - 2a\beta_1 + \beta_1)M^2}{(1 - \beta_1)((1 - a\beta_1) + (\beta_1 + a))} \mathbb{E}[\|\hat{v}_0^{-1/2}\|] \quad \text{where} \quad \Delta f = f(\bar{w}_1) - f(\bar{w}_{T_M+1}) \end{aligned}$$

205 We remark that the bound for our OPT-AMSGrad method matches the complexity bound of  
 206  $\mathcal{O}\left(\sqrt{d/T_M} + 1/T_M\right)$  of [14] for SGD and [43] for AMSGrad method.

### 207 4.3 Checking H1 for a Deep Neural Network:

208 As boundedness assumption H1 is generally hard to verify, we now show, for illustrative purposes,  
 209 that the weights of a fully connected feed forward neural network stay in a bounded set when being  
 210 trained using our method. The activation function for this section will be sigmoid function and we  
 211 use a  $\ell_2$  regularization. We consider a fully connected feed forward neural network with  $L$  layers  
 212 modeled by the function  $\text{MLN}(w, \xi) : \Theta^d \times \mathbb{R}^p \rightarrow \mathbb{R}$ :

$$\text{MLN}(w, \xi) = \sigma \left( w^{(L)} \sigma \left( w^{(L-1)} \dots \sigma \left( w^{(1)} \xi \right) \right) \right) \quad (8)$$

213 where  $w = [w^{(1)}, w^{(2)}, \dots, w^{(L)}]$  is the vector of parameters,  $\xi \in \mathbb{R}^p$  is the input data and  $\sigma$  is the  
 214 sigmoid activation function. We assume a  $p$  dimension input data and a scalar output for simplicity.  
 215 The stochastic objective function (4) reads:

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

where  $\mathcal{L}(\cdot, y)$  is the loss function (can be Huber loss or cross entropy),  $y$  are the true labels and  
 $\lambda > 0$  is the regularization parameter. For any index  $\ell \in [1, L]$  we denote the output of layer  $\ell$  by

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

216 The following Lemma proves that assumption H1 is satisfied with a feed forward neural net (8):

217 **Lemma 4.** Given the multilayer model (8), assume the boundedness of the input data and of the  
 218 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.  
 219 and  $|\mathcal{L}'(\cdot, y)| \leq T$  where  $\mathcal{L}'(\cdot, y)$  denotes its derivative w.r.t. the parameter. Then for each layer  
 220  $\ell \in [1, L]$ , there exist a constant  $A_{(\ell)}$  such that  $\|w^{(\ell)}\| \leq A_{(\ell)}$



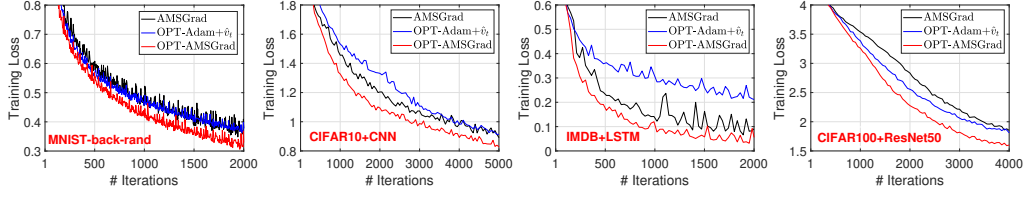


Figure 2: Training loss vs. Number of iterations for fully connected NN, LSTM, CNN and ResNet.

## 5 Numerical Experiments

### 5.1 Gradient Estimation

From the analysis in the previous section, we understand that the choice of the prediction  $m_t$  plays an important role in the convergence of OPTIMISTIC-AMSGRAD. Some classical works in gradient prediction methods include ANDERSON acceleration [37], MINIMAL POLYNOMIAL EXTRAPOLATION [4], REDUCED RANK EXTRAPOLATION [12]. These methods aim at finding a fixed point  $g^*$  and assume that  $\{g_t \in \mathbb{R}^d\}_{t>0}$  has the following linear relation:

$$g_t - g^* = A(g_{t-1} - g^*) + e_t, \quad (9)$$

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 matrix, see [32] for details and results. For our numerical experiments, we run OPT-AMSGRAD using Algorithm 3 to construct the sequence  $\{m_t\}_{t>0}$  and based on estimating the limit of a sequence using the last iterates [3]. Specifically, at iteration  $t$ ,  $m_t$  is obtained by (a) calling Algorithm 3 with

---

#### Algorithm 3 REGULARIZED APPROXIMATE MINIMAL POLYNOMIAL EXTRAPOLATION [32]

---

- 1: **Input:** sequence  $\{g_s \in \mathbb{R}^d\}_{s=0}^{r-1}$ , 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  $g^*$ .
- 

a sequence of past  $r$  gradients,  $\{g_{t-1}, g_{t-2}, \dots, g_{t-r}\}$  as input and (b) setting  $m_t := \sum_{i=0}^{r-1} c_i g_{t-r+i}$  where  $c = [c_0, \dots, c_{r-1}]$  is obtained by Algorithm 3. To see why the output from the extrapolation method may be a 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 (9) 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 is converges to a stationary point, the magnitude of the error will converge to zero.

**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) are parallelizable in the final implementation.

### 5.2 Classification Experiments

In this section, we provide experiments on classification tasks with various neural network architectures and datasets to demonstrate the effectiveness of OPT-AMSGRAD.

**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 [31]. The other benchmark method is the OPTIMISTIC-ADAM+ $\hat{v}_t$  [8], which details are reported to the Supplementary Material. 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  $\eta$  of AMSGRAD for a fair evaluation of the optimistic step. 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 experiments in the supplementary material). The algorithms are initialized at the same point and the results are averaged over 5 repetitions.

**Datasets.** We compare different algorithms on *MNIST*, *CIFAR10*, *CIFAR100*, and *IMDB* datasets. For *MNIST*, we use two noisy variants called *MNIST-back-rand* and *MNIST-back-image* from [20] ( $n = 12\,000$ ), *CIFAR10* and *CIFAR100* ( $n = 50\,000$ ) and *IMDB* ( $n = 25\,000$ ).

**Network architecture.** We adopt a multi-layer fully connected neural network with hidden layers of 200 then 100 neurons (using ReLU activations and Softmax output) on *MNIST* variants. For *CIFAR* datasets, we adopt ALL-CNN network proposed by [33], built with convolutional blocks and dropout layers. In addition, we also apply residual networks, Resnet-18 and Resnet-50 [18], which have achieved state-of-the-art results. For the texture *IMDB* dataset, we consider a Long-Short Term Memory (LSTM) network [13] including 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 2. 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 3, 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 objective function value, and (2) in these three applications, the proposed OPT-AMSGRAD also outperforms the competing methods in terms of test accuracy.

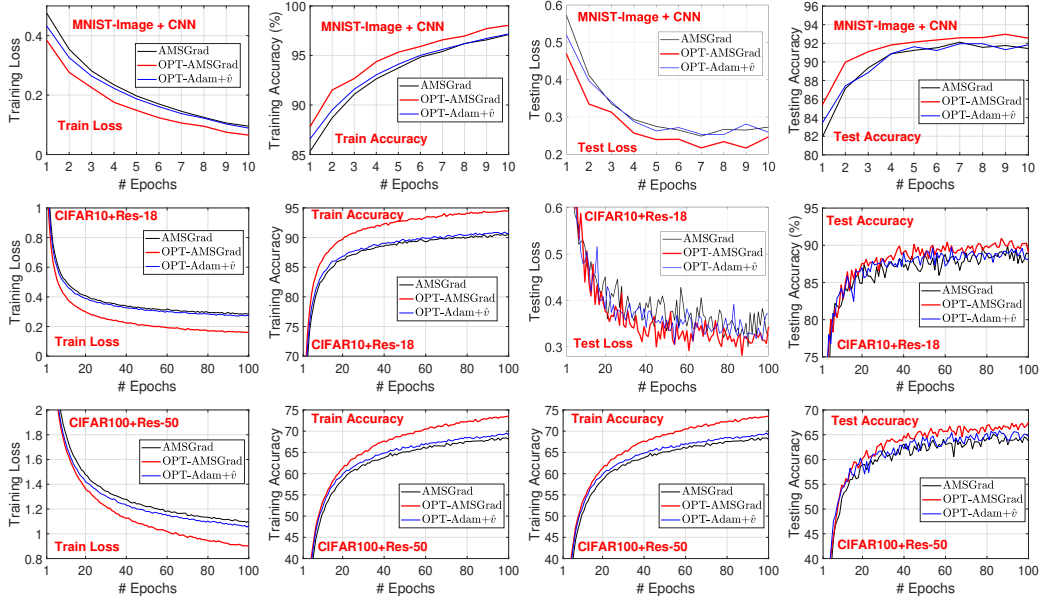


Figure 3: *MNIST-back-image* + CNN, *CIFAR10* + Res-18 and *CIFAR100* + Res-50 . We compare three methods in terms of training (cross-entropy) loss and accuracy, testing loss and accuracy.

## 6 Conclusion

In this paper, we propose OPT-AMSGRAD, which combines optimistic online learning and AMSGRAD to improve sample efficiency and accelerate the process of training, in particular for deep neural networks. With a good gradient prediction process, we demonstrate that the regret can be smaller than that of standard AMSGRAD. Experiments on various deep learning problems demonstrate the effectiveness of the proposed method in accelerating the empirical risk minimization procedure.



## References

- [1] J. Abernethy, K. A. Lai, K. Y. Levy, and J.-K. Wang. Faster rates for convex-concave games. *COLT*, 2018.
- [2] N. Agarwal, B. Bullins, X. Chen, E. Hazan, K. Singh, C. Zhang, and Y. Zhang. Efficient full-matrix adaptive regularization. *ICML*, 2019.
- [3] C. Brezinski and M. R. Zaglia. Extrapolation methods: theory and practice. *Elsevier*, 2013.
- [4] S. Cabay and L. Jackson. A polynomial extrapolation method for finding limits and antilimits of vector sequences. *SIAM Journal on Numerical Analysis*, 1976.
- [5] X. Chen, S. Liu, R. Sun, and M. Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. *ICLR*, 2019.
- [6] Z. Chen, Z. Yuan, J. Yi, B. Zhou, E. Chen, and T. Yang. Universal stagewise learning for non-convex problems with convergence on averaged solutions. *ICLR*, 2019.
- [7] C.-K. Chiang, T. Yang, C.-J. Lee, M. Mahdavi, C.-J. Lu, R. Jin, and S. Zhu. Online optimization with gradual variations. *COLT*, 2012.
- [8] C. Daskalakis, A. Ilyas, V. Syrgkanis, and H. Zeng. Training gans with optimism. *ICLR*, 2018.
- [9] A. Défossez, L. Bottou, F. Bach, and N. Usunier. On the convergence of adam and adagrad. *arXiv preprint arXiv:2003.02395*, 2020.
- [10] T. Dozat. Incorporating nesterov momentum into adam. *ICLR (Workshop Track)*, 2016.
- [11] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research (JMLR)*, 2011.
- [12] R. Eddy. Extrapolating to the limit of a vector sequence. *Information linkage between applied mathematics and industry*, Elsevier, 1979.
- [13] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with lstm. 1999.
- [14] S. Ghadimi and G. Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.
- [15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *NIPS*, 2014.
- [16] A. Graves, A. rahman Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. *ICASSP*, 2013.
- [17] E. Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2016.
- [18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, 2016.
- [19] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.
- [20] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. *ICML*, 2007.
- [21] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *NIPS*, 2017.
- [22] X. Li and F. Orabona. On the convergence of stochastic gradient descent with adaptive step-sizes. *AISTAT*, 2019.

- [23] H. B. McMahan and M. J. Streeter. Adaptive bound optimization for online convex optimization. *COLT*, 2010.
- [24] P. Mertikopoulos, B. Lecouat, H. Zenati, C.-S. Foo, V. Chandrasekhar, and G. Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. *arXiv preprint arXiv:1807.02629*, 2018.
- [25] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *NIPS (Deep Learning Workshop)*, 2013.
- [26] M. Mohri and S. Yang. Accelerating optimization via adaptive prediction. *AISTATS*, 2016.
- [27] Y. Nesterov. Introductory lectures on convex optimization: A basic course. *Springer*, 2004.
- [28] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *Mathematics and Mathematical Physics*, 1964.
- [29] A. Rakhlin and K. Sridharan. Optimization, learning, and games with predictable sequences. *NIPS*, 2013.
- [30] S. Rakhlin and K. Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074, 2013.
- [31] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. *ICLR*, 2018.
- [32] D. Scieur, A. d’Aspremont, and F. Bach. Regularized nonlinear acceleration. *NIPS*, 2016.
- [33] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *ICLR*, 2015.
- [34] V. Syrgkanis, A. Agarwal, H. Luo, and R. E. Schapire. Fast convergence of regularized learning in games. *NIPS*, 2015.
- [35] T. Tieleman and G. Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 2012.
- [36] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. 2008.
- [37] H. F. Walker and P. Ni. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, 2011.
- [38] R. Ward, X. Wu, and L. Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes, from any initialization. *ICML*, 2019.
- [39] Y. Yan, T. Yang, Z. Li, Q. Lin, and Y. Yang. A unified analysis of stochastic momentum methods for deep learning. *arXiv preprint arXiv:1808.10396*, 2018.
- [40] M. Zaheer, S. Reddi, D. Sachan, S. Kale, and S. Kumar. Adaptive methods for nonconvex optimization. *NeurIPS*, 2018.
- [41] M. D. Zeiler. Adadelta: An adaptive learning rate method. *arXiv:1212.5701*, 2012.
- [42] D. Zhou, Y. Tang, Z. Yang, Y. Cao, and Q. Gu. On the convergence of adaptive gradient methods for nonconvex optimization. *arXiv:1808.05671*, 2018.
- [43] D. Zhou, Y. Tang, Z. Yang, Y. Cao, and Q. Gu. On the convergence of adaptive gradient methods for nonconvex optimization. *arXiv preprint arXiv:1808.05671*, 2018.
- [44] F. Zou and L. Shen. On the convergence of adagrad with momentum for training deep neural networks. *arXiv:1808.03408*, 2018.

## A Proof of Theorem 1

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

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

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  $D_\infty^2$  is the diameter of the bounded set  $\Theta$ . The result holds for any benchmark  $w^* \in \Theta$  and any step size sequence  $\{\eta_t\}_{t>0}$ .

**Proof** Beforehand, note:

$$\begin{aligned} \tilde{g}_t &= \beta_1 \theta_{t-1} + (1 - \beta_1) g_t \\ \tilde{m}_{t+1} &= \beta_1 \theta_{t-1} + (1 - \beta_1) m_{t+1} \end{aligned} \quad (10)$$

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

$$\begin{aligned} \text{Regret}_T &:= \sum_{t=1}^T \ell_t(w_t) - \min_{w \in \Theta} \sum_{t=1}^T \ell_t(w) \\ &\leq \sum_{t=1}^T \langle w_t - w^*, \nabla \ell_t(w_t) \rangle \\ &= \sum_{t=1}^T \langle w_t - \tilde{w}_{t+1}, g_t - \tilde{m}_t \rangle + \langle w_t - \tilde{w}_{t+1}, \tilde{m}_t \rangle + \langle \tilde{w}_{t+1} - w^*, \tilde{g}_t \rangle + \langle \tilde{w}_{t+1} - w^*, g_t - \tilde{g}_t \rangle. \end{aligned} \quad (11)$$

Recall the notation  $\psi_t(x)$  and the Bregman divergence  $B_{\psi_t}(u, v)$  we defined in the beginning of this section. Now we are going to exploit a useful inequality (which appears in e.g., [36]); for any update of the form  $\hat{w} = \arg \min_{w \in \Theta} \langle w, \theta \rangle + B_\psi(w, v)$ , it holds that

$$\langle \hat{w} - u, \theta \rangle \leq B_\psi(u, v) - B_\psi(u, \hat{w}) - B_\psi(\hat{w}, v) \quad \text{for any } u \in \Theta. \quad (12)$$

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

$$\tilde{w}_{t+1} = \arg \min_{w \in \Theta} \eta_t \langle w, \tilde{g}_t \rangle + B_{\psi_t}(w, \tilde{w}_t), \quad (13)$$

By using (12) for (13) with  $\hat{w} = \tilde{w}_{t+1}$  (the output of the minimization problem),  $u = w^*$  and  $v = \tilde{w}_t$ , we have

$$\langle \tilde{w}_{t+1} - w^*, \tilde{g}_t \rangle \leq \frac{1}{\eta_t} [B_{\psi_t}(w^*, \tilde{w}_t) - B_{\psi_t}(w^*, \tilde{w}_{t+1}) - B_{\psi_t}(\tilde{w}_{t+1}, \tilde{w}_t)]. \quad (14)$$

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

$$w_{t+1} = \arg \min_{w \in \Theta} \eta_{t+1} \langle w, \tilde{m}_{t+1} \rangle + B_{\psi_t}(w, \tilde{w}_{t+1}). \quad (15)$$

and, by using (12) for (15) (written at iteration  $t$ ), with  $\hat{w} = w_t$  (the output of the minimization problem),  $u = \tilde{w}_{t+1}$  and  $v = \tilde{w}_t$ , we have

$$\langle w_t - \tilde{w}_{t+1}, \tilde{m}_t \rangle \leq \frac{1}{\eta_t} [B_{\psi_{t-1}}(\tilde{w}_{t+1}, \tilde{w}_t) - B_{\psi_{t-1}}(\tilde{w}_{t+1}, w_t) - B_{\psi_{t-1}}(w_t, \tilde{w}_t)], \quad (16)$$

By (11), (14), and (16), we obtain

$$\begin{aligned} \mathcal{R}_T &\stackrel{(11)}{\leq} \sum_{t=1}^T \langle w_t - \tilde{w}_{t+1}, g_t - \tilde{m}_t \rangle + \langle w_t - \tilde{w}_{t+1}, \tilde{m}_t \rangle + \langle \tilde{w}_{t+1} - w^*, \tilde{g}_t \rangle + \langle \tilde{w}_{t+1} - w^*, g_t - \tilde{g}_t \rangle \\ &\stackrel{(14),(16)}{\leq} \sum_{t=1}^T \|w_t - \tilde{w}_{t+1}\|_{\psi_{t-1}} \|g_t - \tilde{m}_t\|_{\psi_{t-1}^*} + \|\tilde{w}_{t+1} - w^*\|_{\psi_{t-1}} \|g_t - \tilde{g}_t\|_{\psi_{t-1}^*} \\ &\quad + \frac{1}{\eta_t} [B_{\psi_{t-1}}(\tilde{w}_{t+1}, \tilde{w}_t) - B_{\psi_{t-1}}(\tilde{w}_{t+1}, w_t) - B_{\psi_{t-1}}(w_t, \tilde{w}_t) + B_{\psi_t}(w^*, \tilde{w}_t) - B_{\psi_t}(w^*, \tilde{w}_{t+1}) - B_{\psi_t}(\tilde{w}_{t+1}, \tilde{w}_t)], \end{aligned} \quad (17)$$

381 which is further bounded by

$$\begin{aligned} \mathcal{R}_T \leq & \sum_{t=1}^T \left\{ \frac{1}{2\eta_t} \|w_t - \tilde{w}_{t+1}\|_{\psi_{t-1}}^2 + \frac{\eta_t}{2} \|g_t - m_t\|_{\psi_{t-1}^*}^2 + \|\tilde{w}_{t+1} - w^*\|_{\psi_{t-1}} \|g_t - \tilde{g}_t\|_{\psi_{t-1}^*} \right. \\ & \left. + \frac{1}{\eta_t} \left( \underbrace{B_{\psi_{t-1}}(\tilde{w}_{t+1}, \tilde{w}_t) - B_{\psi_t}(\tilde{w}_{t+1}, \tilde{w}_t)}_{A_1} - \frac{1}{2} \|\tilde{w}_{t+1} - w_t\|_{\psi_{t-1}}^2 + \underbrace{B_{\psi_t}(w^*, \tilde{w}_t) - B_{\psi_t}(w^*, \tilde{w}_{t+1})}_{A_2} \right) \right\}, \end{aligned} \quad (18)$$

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

385 To proceed, notice that

$$A_1 = B_{\psi_{t-1}}(\tilde{w}_{t+1}, \tilde{w}_t) - B_{\psi_t}(\tilde{w}_{t+1}, \tilde{w}_t) = \langle \tilde{w}_{t+1} - \tilde{w}_t, \text{diag}(\hat{v}_{t-1}^{1/2} - \hat{v}_t^{1/2})(\tilde{w}_{t+1} - \tilde{w}_t) \rangle \leq 0, \quad (19)$$

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

$$\begin{aligned} A_2 &= B_{\psi_t}(w^*, \tilde{w}_t) - B_{\psi_t}(w^*, \tilde{w}_{t+1}) = \langle w^* - \tilde{w}_{t+1}, \text{diag}(\hat{v}_{t+1}^{1/2} - \hat{v}_t^{1/2})(w^* - \tilde{w}_{t+1}) \rangle \\ &\leq (\max_i (w^*[i] - \tilde{w}_{t+1}[i])^2) \cdot \left( \sum_{i=1}^d \hat{v}_{t+1}^{1/2}[i] - \hat{v}_t^{1/2}[i] \right) \end{aligned} \quad (20)$$

387 Therefore, by (18),(20),(19), we have

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

388 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  
 389 proof.

390 □

## 391 B Proof of Corollary 1

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

$$\mathcal{R}_T \leq \frac{B_{\psi_1}(w^*, \tilde{w}_1)}{\eta_1} + \frac{\eta\sqrt{1+\log T}}{\sqrt{1-\beta_2}} \sum_{i=1}^d \|(g - m)_{1:T}[i]\|_2 + \frac{D_\infty^2}{\eta_{\min}} \sum_{i=1}^d \left[ (1 - \beta_2) \sum_{s=1}^T \beta_2^{T-s} g_s[i]^2 \right]^{1/2},$$

394 where  $g_t := \nabla \ell_t(w_t)$  and  $\eta_{\min} := \min_t \eta_t$ .

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

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

396 The second term reads:

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

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

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

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

400

□

## 401 C Proofs of Auxiliary Lemmas

### 402 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:

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

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

$$\|\theta_{t+1}\| = \|\beta_1 \theta_t + (1 - \beta_1) g_{t+1}\| \leq \beta_1 \|\theta_t\| + (1 - \beta_1) \|g_{t+1}\| \leq M \quad (21)$$

404 Using the same induction reasoning we prove that

$$\|\hat{v}_{t+1}\| = \|\beta_2 \hat{v}_t + (1 - \beta_2) g_{t+1}^2\| \leq \beta_2 \|\hat{v}_t\| + (1 - \beta_1) \|g_{t+1}^2\| \leq M^2 \quad (22)$$

405

□

### 406 C.2 Proof of Lemma 2

407 **Lemma.** Assume a strictly positive and non increasing sequence of stepsizes  $\{\eta_t\}_{t>0}$ ,  $\beta \in [0, 1]$ , then  
408 the following holds:

$$\bar{w}_{t+1} - \bar{w}_t \leq \frac{\beta_1}{1 - \beta_1} \tilde{\theta}_{t-1} \left[ \eta_{t-1} \hat{v}_{t-1}^{-1/2} - \eta_t \hat{v}_t^{-1/2} \right] - \eta_t \hat{v}_t^{-1/2} \tilde{g}_t, \quad (23)$$

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

410 **Proof** By definition (6) and using the Algorithm updates, we have:

$$\begin{aligned} \bar{w}_{t+1} - \bar{w}_t &= \frac{1}{1 - \beta_1} (w_{t+1} - \tilde{w}_t) - \frac{\beta_1}{1 - \beta_1} (w_t - \tilde{w}_{t-1}) \\ &= -\frac{1}{1 - \beta_1} \eta_t \hat{v}_t^{-1/2} (\theta_t + h_{t+1}) + \frac{\beta_1}{1 - \beta_1} \eta_{t-1} \hat{v}_{t-1}^{-1/2} (\theta_{t-1} + h_t) \\ &= -\frac{1}{1 - \beta_1} \eta_t \hat{v}_t^{-1/2} (\theta_t + \beta_1 \theta_{t-1}) - \frac{1}{1 - \beta_1} \eta_t \hat{v}_t^{-1/2} (1 - \beta_1) m_{t+1} \\ &\quad + \frac{\beta_1}{1 - \beta_1} \eta_{t-1} \hat{v}_{t-1}^{-1/2} (\theta_{t-1} + \beta_1 \theta_{t-2}) + \frac{\beta_1}{1 - \beta_1} \eta_{t-1} \hat{v}_{t-1}^{-1/2} (1 - \beta_1) m_t \end{aligned} \quad (24)$$

411 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  $\tilde{\theta}_t = \beta_1 \tilde{\theta}_{t-1} + (1 -$   
412  $\beta_1)(g_t + \beta_1 g_{t-1})$ .

$$\bar{w}_{t+1} - \bar{w}_t \leq \frac{\beta_1}{1 - \beta_1} \tilde{\theta}_{t-1} \left[ \eta_{t-1} \hat{v}_{t-1}^{-1/2} - \eta_t \hat{v}_t^{-1/2} \right] - \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \quad (25)$$

413

□

### 414 C.3 Proof of Lemma 3

415 **Lemma.** Assume H4, a strictly positive and a sequence of constant stepsizes  $\{\eta_t\}_{t>0}$ ,  $\beta \in [0, 1]$ , then  
416 the following holds:

$$\sum_{t=1}^{T_M} \eta_t^2 \mathbb{E} \left[ \left\| \hat{v}_t^{-1/2} \theta_t \right\|_2^2 \right] \leq \frac{\eta^2 d T_M (1 - \beta_1)}{(1 - \beta_2)(1 - \gamma)} \quad (26)$$



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

$$\begin{aligned} \eta_t^2 \mathbb{E} \left[ \left\| \hat{v}_t^{-1/2} \theta_t \right\|_2^2 \right] &= \eta_t^2 \mathbb{E} \left[ \sum_{p=1}^d \frac{\theta_{t,p}^2}{\hat{v}_{t,p}} \right] \\ &\leq \eta_t^2 \mathbb{E} \left[ \sum_{i=1}^d \frac{\theta_{t,p}^2}{v_{t,p}} \right] \\ &\leq \eta_t^2 \mathbb{E} \left[ \sum_{i=1}^d \frac{(\sum_{r=1}^t (1 - \beta_1) \beta_1^{t-r} g_{r,p})^2}{\sum_{r=1}^t (1 - \beta_2) \beta_2^{t-r} g_{r,p}^2} \right] \end{aligned} \quad (27)$$

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

$$\begin{aligned} \eta_t^2 \mathbb{E} \left[ \left\| \hat{v}_t^{-1/2} \theta_t \right\|_2^2 \right] &\leq \frac{\eta_t^2 (1 - \beta_1)^2}{1 - \beta_2} \mathbb{E} \left[ \sum_{i=1}^d \frac{(\sum_{r=1}^t \beta_1^{t-r} g_{r,p})^2}{\sum_{r=1}^t \beta_2^{t-r} g_{r,p}^2} \right] \\ &\stackrel{(a)}{\leq} \frac{\eta_t^2 (1 - \beta_1)}{1 - \beta_2} \mathbb{E} \left[ \sum_{i=1}^d \frac{\sum_{r=1}^t \beta_1^{t-r} g_{r,p}^2}{\sum_{r=1}^t \beta_2^{t-r} g_{r,p}^2} \right] \\ &\leq \frac{\eta_t^2 (1 - \beta_1)}{1 - \beta_2} \mathbb{E} \left[ \sum_{i=1}^d \sum_{r=1}^t \gamma^{t-r} \right] = \frac{\eta_t^2 d (1 - \beta_1)}{1 - \beta_2} \mathbb{E} \left[ \sum_{r=1}^t \gamma^{t-r} \right] \end{aligned} \quad (28)$$

420 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:

$$\begin{aligned} \sum_{t=1}^{T_M} \eta_t^2 \mathbb{E} \left[ \left\| \hat{v}_t^{-1/2} \theta_t \right\|_2^2 \right] &\leq \frac{\eta_t^2 d (1 - \beta_1)}{1 - \beta_2} \mathbb{E} \left[ \sum_{t=1}^{T_M} \sum_{r=1}^t \gamma^{t-r} \right] \\ &\leq \frac{\eta^2 d T (1 - \beta_1)}{1 - \beta_2} \mathbb{E} \left[ \sum_{t=1}^t \gamma^{t-r} \right] \\ &\leq \frac{\eta^2 d T (1 - \beta_1)}{(1 - \beta_2)(1 - \gamma)} \end{aligned} \quad (29)$$

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

## 422 D Proof of Theorem 2

423 **Theorem.** Assume H2-H4,  $(\beta_1, \beta_2) \in [0, 1]$  and a sequence of decreasing stepsizes  $\{\eta_t\}_{t>0}$ , then  
 424 the following result holds:

$$\mathbb{E} [\|\nabla f(w_T)\|^2] \leq \tilde{C}_1 \sqrt{\frac{d}{T_M}} + \tilde{C}_2 \frac{1}{T_M} \quad (30)$$

425 where  $T$  is a random termination number distributed according (5) and the constants are defined as  
 426 follows:

$$\begin{aligned} \tilde{C}_1 &= C_1 + \frac{M}{(1 - a\beta_1) + (\beta_1 + a)} \left[ \frac{a(1 - \beta_1)^2}{1 - \beta_2} + 2L \frac{1}{1 - \beta_2} \right] \\ C_1 &= \frac{M}{(1 - a\beta_1) + (\beta_1 + a)} \Delta f + \frac{4L \left( \frac{\beta_1}{1 - \beta_1} \right)^2 M}{(1 - a\beta_1) + (\beta_1 + a)} \frac{(1 + \beta_1^2)(1 - \beta_1)}{(1 - \beta_2)(1 - \gamma)} \\ \tilde{C}_2 &= \frac{M}{(1 - \beta_1)((1 - a\beta_1) + (\beta_1 + a))} \tilde{M}^2 \mathbb{E} \left[ \left\| \hat{v}_0^{-1/2} \right\| \right] \end{aligned} \quad (31)$$

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

$$\begin{aligned} f(\bar{w}_{t+1}) &\leq f(\bar{w}_t) + \nabla f(\bar{w}_t)^\top (\bar{w}_{t+1} - \bar{w}_t) + \frac{L}{2} \|\bar{w}_{t+1} - \bar{w}_t\|^2 \\ &\leq f(\bar{w}_t) + \underbrace{\nabla f(w_t)^\top (\bar{w}_{t+1} - \bar{w}_t)}_A + \underbrace{(\nabla f(\bar{w}_t) - \nabla f(w_t))^\top (\bar{w}_{t+1} - \bar{w}_t)}_B + \frac{L}{2} \|\bar{w}_{t+1} - \bar{w}_t\| \end{aligned} \quad (32)$$

428 **Term A.** Using Lemma 2, we have that:

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

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

$$\nabla f(w_t)^\top (\bar{w}_{t+1} - \bar{w}_t) \leq \frac{\beta_1(1 + \beta_1)}{1 - \beta_1} \mathbf{M}^2 \left[ \left\| \eta_{t-1} \hat{v}_{t-1}^{-1/2} \right\| - \left\| \eta_t \hat{v}_t^{-1/2} \right\| \right] - \nabla f(w_t)^\top \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \quad (34)$$

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

$$\begin{aligned} -\nabla f(w_t)^\top \eta_t \hat{v}_t^{-1/2} \tilde{g}_t &= -\nabla f(w_t)^\top \eta_{t-1} \hat{v}_{t-1}^{-1/2} \bar{g}_t - \nabla f(w_t)^\top \left[ \eta_t \hat{v}_t^{-1/2} - \eta_{t-1} \hat{v}_{t-1}^{-1/2} \right] \bar{g}_t \\ &\quad - \nabla f(w_t)^\top \eta_{t-1} \hat{v}_{t-1}^{-1/2} (\beta_1 g_{t-1} + m_{t+1}) \\ &\leq -\nabla f(w_t)^\top \eta_{t-1} \hat{v}_{t-1}^{-1/2} \bar{g}_t + (1 - a\beta_1) \mathbf{M}^2 \left[ \left\| \eta_{t-1} \hat{v}_{t-1}^{-1/2} \right\| - \left\| \eta_t \hat{v}_t^{-1/2} \right\| \right] \\ &\quad - \nabla f(w_t)^\top \eta_t \hat{v}_t^{-1/2} (\beta_1 g_{t-1} + m_{t+1}) \end{aligned} \quad (35)$$

433 using Lemma 1 on  $\|g_t\|$  and where that  $\tilde{g}_t = \bar{g}_t + \beta_1 g_{t-1} + m_{t+1} = g_t - \beta_1 m_t + \beta_1 g_{t-1} + m_{t+1}$ .  
434 Plugging (35) into (34) yields:

$$\begin{aligned} &\nabla f(w_t)^\top (\bar{w}_{t+1} - \bar{w}_t) \\ &\leq -\nabla f(w_t)^\top \eta_{t-1} \hat{v}_{t-1}^{-1/2} \bar{g}_t + \frac{1}{1 - \beta_1} (a\beta_1^2 - 2a\beta_1 + \beta_1) \mathbf{M}^2 \left[ \left\| \eta_{t-1} \hat{v}_{t-1}^{-1/2} \right\| - \left\| \eta_t \hat{v}_t^{-1/2} \right\| \right] \\ &\quad - \nabla f(w_t)^\top \eta_t \hat{v}_t^{-1/2} (\beta_1 g_{t-1} + m_{t+1}) \end{aligned} \quad (36)$$

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

$$(\nabla f(\bar{w}_t) - \nabla f(w_t))^\top (\bar{w}_{t+1} - \bar{w}_t) \leq \|\nabla f(\bar{w}_t) - \nabla f(w_t)\| \|\bar{w}_{t+1} - \bar{w}_t\| \quad (37)$$

436 Using smoothness assumption H2:

$$\begin{aligned} \|\nabla f(\bar{w}_t) - \nabla f(w_t)\| &\leq L \|\bar{w}_t - w_t\| \\ &\leq L \frac{\beta_1}{1 - \beta_1} \|w_t - \tilde{w}_{t-1}\| \end{aligned} \quad (38)$$

437 By Lemma 2 we also have:

$$\begin{aligned} \bar{w}_{t+1} - \bar{w}_t &= \frac{\beta_1}{1 - \beta_1} \tilde{\theta}_{t-1} \left[ \eta_{t-1} \hat{v}_{t-1}^{-1/2} - \eta_t \hat{v}_t^{-1/2} \right] - \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \\ &= \frac{\beta_1}{1 - \beta_1} \tilde{\theta}_{t-1} \eta_{t-1} \hat{v}_{t-1}^{-1/2} \left[ I - (\eta_t \hat{v}_t^{-1/2})(\eta_{t-1} \hat{v}_{t-1}^{-1/2})^{-1} \right] - \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \\ &= \frac{\beta_1}{1 - \beta_1} \left[ I - (\eta_t \hat{v}_t^{-1/2})(\eta_{t-1} \hat{v}_{t-1}^{-1/2})^{-1} \right] (\tilde{w}_{t-1} - w_t) - \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \end{aligned} \quad (39)$$

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  $\left\|I - (\eta_t\hat{v}_t^{-1/2})(\eta_{t-1}\hat{v}_{t-1}^{-1/2})^{-1}\right\| \leq 1$  due to the decreasing stepsize and the construction of  $\hat{v}_t$  and using CS inequality yield:

$$\|\bar{w}_{t+1} - \bar{w}_t\| \leq \frac{\beta_1}{1-\beta_1} \|\tilde{w}_{t-1} - w_t\| + \left\|\eta_t\hat{v}_t^{-1/2}\tilde{g}_t\right\| \quad (40)$$

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 (38) and (40) into (37) returns:

$$\begin{aligned} (\nabla f(\bar{w}_t) - \nabla f(w_t))^\top (\bar{w}_{t+1} - \bar{w}_t) &\leq L \frac{\beta_1}{1-\beta_1} \left\|\eta_t\hat{v}_t^{-1/2}\tilde{g}_t\right\| \|w_t - \tilde{w}_{t-1}\| \\ &\quad + L \left(\frac{\beta_1}{1-\beta_1}\right)^2 \|\tilde{w}_{t-1} - w_t\|^2 \end{aligned} \quad (41)$$

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

$$(\nabla f(\bar{w}_t) - \nabla f(w_t))^\top (\bar{w}_{t+1} - \bar{w}_t) \leq L \left\|\eta_t\hat{v}_t^{-1/2}\tilde{g}_t\right\|^2 + 2L \left(\frac{\beta_1}{1-\beta_1}\right)^2 \|\tilde{w}_{t-1} - w_t\|^2 \quad (42)$$

The last term  $\frac{L}{2} \|\bar{w}_{t+1} - \bar{w}_t\|^2$  can be upper bounded using (40):

$$\begin{aligned} \frac{L}{2} \|\bar{w}_{t+1} - \bar{w}_t\|^2 &\leq \frac{L}{2} \left[ \frac{\beta_1}{1-\beta_1} \|\tilde{w}_{t-1} - w_t\| + \left\|\eta_t\hat{v}_t^{-1/2}\tilde{g}_t\right\| \right]^2 \\ &\leq L \left\|\eta_t\hat{v}_t^{-1/2}\tilde{g}_t\right\|^2 + 2L \left(\frac{\beta_1}{1-\beta_1}\right)^2 \|\tilde{w}_{t-1} - w_t\|^2 \end{aligned} \quad (43)$$

Plugging (36), (42) and (43) into (32) and taking the expectations on both sides give:

$$\begin{aligned} &\mathbb{E} \left[ f(\bar{w}_{t+1}) + \frac{1}{1-\beta_1} \tilde{M}^2 \left\|\eta_t\hat{v}_t^{-1/2}\right\| - \left( f(\bar{w}_t) + \frac{1}{1-\beta_1} \tilde{M}^2 \left\|\eta_{t-1}\hat{v}_{t-1}^{-1/2}\right\| \right) \right] \\ &\leq \mathbb{E} \left[ -\nabla f(w_t)^\top \eta_{t-1}\hat{v}_{t-1}^{-1/2}\tilde{g}_t - \nabla f(w_t)^\top \eta_t\hat{v}_t^{-1/2}(\beta_1 g_{t-1} + m_{t+1}) \right] \\ &\quad + \mathbb{E} \left[ 2L \left\|\eta_t\hat{v}_t^{-1/2}\tilde{g}_t\right\|^2 + 4L \left(\frac{\beta_1}{1-\beta_1}\right)^2 \|\tilde{w}_{t-1} - w_t\|^2 \right] \end{aligned} \quad (44)$$

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

$$\begin{aligned} \mathbb{E} [\nabla f(w_t)^\top \tilde{g}_t] &= \mathbb{E} [\nabla f(w_t)^\top (g_t - \beta_1 m_t)] \\ &= (1 - a\beta_1) \|\nabla f(w_t)\|^2 \end{aligned} \quad (45)$$

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

$$\begin{aligned} &\frac{1}{M} \sum_{t=1}^{T_M} ((1-a\beta_1)\eta_{t-1} + (\beta_1+a)\eta_t) \|\nabla f(w_t)\|^2 \leq \\ &\mathbb{E} \left[ f(\bar{w}_1) + \frac{1}{1-\beta_1} \tilde{M}^2 \left\|\eta_0\hat{v}_0^{-1/2}\right\| - \left( f(\bar{w}_{T_M+1}) + \frac{1}{1-\beta_1} \tilde{M}^2 \left\|\eta_{T_M}\hat{v}_{T_M}^{-1/2}\right\| \right) \right] \\ &\quad + 2L \sum_{t=1}^{T_M} \mathbb{E} \left[ \left\|\eta_t\hat{v}_t^{-1/2}\tilde{g}_t\right\|^2 \right] + 4L \left(\frac{\beta_1}{1-\beta_1}\right)^2 \sum_{t=1}^{T_M} \mathbb{E} [\|\tilde{w}_{t-1} - w_t\|^2] \\ &\leq \mathbb{E} \left[ \Delta f + \frac{1}{1-\beta_1} \tilde{M}^2 \left\|\eta_0\hat{v}_0^{-1/2}\right\| \right] + 2L \sum_{t=1}^{T_M} \mathbb{E} \left[ \left\|\eta_t\hat{v}_t^{-1/2}\tilde{g}_t\right\|^2 \right] + 4L \left(\frac{\beta_1}{1-\beta_1}\right)^2 \sum_{t=1}^{T_M} \mathbb{E} [\|\tilde{w}_{t-1} - w_t\|^2] \end{aligned} \quad (46)$$

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

$$\begin{aligned}\|\tilde{w}_{t-1} - w_t\|^2 &= \left\| \eta_{t-1} \hat{v}_{t-1}^{-1/2} (\theta_{t-1} + h_t) \right\|^2 \\ &= \left\| \eta_{t-1} \hat{v}_{t-1}^{-1/2} (\theta_{t-1} + \beta_1 \theta_{t-2} + (1 - \beta_1) m_t) \right\|^2 \\ &\leq \left\| \eta_{t-1} \hat{v}_{t-1}^{-1/2} \theta_{t-1} \right\|^2 + \left\| \eta_{t-2} \hat{v}_{t-2}^{-1/2} \beta_1 \theta_{t-2} \right\|^2 + (1 - \beta_1)^2 \left\| \eta_{t-1} \hat{v}_{t-1}^{-1/2} m_t \right\|^2\end{aligned}\tag{47}$$

450 Using Lemma 3 we have

$$\begin{aligned}&\sum_{t=1}^{T_M} \mathbb{E} \left[ \|\tilde{w}_{t-1} - w_t\|^2 \right] \\ &\leq (1 + \beta_1^2) \frac{\eta^2 d T_M (1 - \beta_1)}{(1 - \beta_2)(1 - \gamma)} + (1 - \beta_1)^2 \sum_{t=1}^{T_M} \mathbb{E} \left[ \left\| \eta_{t-1} \hat{v}_{t-1}^{-1/2} m_t \right\|^2 \right]\end{aligned}\tag{48}$$

451 And thus, setting the learning rate to a constant value  $\eta$  and injecting in (46) yields:

$$\begin{aligned}\mathbb{E} [\|\nabla f(w_T)\|^2] &= \frac{1}{\sum_{j=1}^{T_M} \eta_j} \sum_{t=1}^{T_M} \eta_t \|\nabla f(w_t)\|^2 \\ &\leq \frac{M}{(1 - a\beta_1) + (\beta_1 + a)} \frac{1}{\sum_{j=1}^{T_M} \eta_j} \mathbb{E} \left[ \Delta f + \frac{1}{1 - \beta_1} \tilde{M}^2 \left\| \eta_0 \hat{v}_0^{-1/2} \right\| \right] \\ &\quad + \frac{4L \left( \frac{\beta_1}{1 - \beta_1} \right)^2 M}{(1 - a\beta_1) + (\beta_1 + a)} \frac{1}{\sum_{j=1}^{T_M} \eta_j} (1 + \beta_1^2) \frac{\eta^2 d T_M (1 - \beta_1)}{(1 - \beta_2)(1 - \gamma)} \\ &\quad + \frac{M}{(1 - a\beta_1) + (\beta_1 + a)} \frac{1}{\sum_{j=1}^{T_M} \eta_j} (1 - \beta_1)^2 \sum_{t=1}^{T_M} \mathbb{E} \left[ \left\| \eta_{t-1} \hat{v}_{t-1}^{-1/2} m_t \right\|^2 \right] \\ &\quad + \frac{2LM}{(1 - a\beta_1) + (\beta_1 + a)} \frac{1}{\sum_{j=1}^{T_M} \eta_j} \sum_{t=1}^{T_M} \mathbb{E} \left[ \left\| \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \right\|^2 \right]\end{aligned}\tag{49}$$

452 where  $T$  is a random termination number distributed according (5). Setting the stepsize to  $\eta = \frac{1}{\sqrt{dT_M}}$   
 453 yields :

$$\begin{aligned}\mathbb{E} [\|\nabla f(w_T)\|^2] &\leq C_1 \sqrt{\frac{d}{T_M}} + C_2 \frac{1}{T_M} \\ &\quad + D_1 \frac{\eta}{T_M} \sum_{t=1}^{T_M} \mathbb{E} \left[ \left\| \hat{v}_{t-1}^{-1/2} m_t \right\|^2 \right] + D_2 \frac{\eta}{T_M} \sum_{t=1}^{T_M} \mathbb{E} \left[ \left\| \hat{v}_{t-1}^{-1/2} \tilde{g}_t \right\|^2 \right]\end{aligned}\tag{50}$$

454 where

$$\begin{aligned}C_1 &= \frac{M}{(1 - a\beta_1) + (\beta_1 + a)} \Delta f + \frac{4L \left( \frac{\beta_1}{1 - \beta_1} \right)^2 M}{(1 - a\beta_1) + (\beta_1 + a)} \frac{(1 + \beta_1^2)(1 - \beta_1)}{(1 - \beta_2)(1 - \gamma)} \\ C_2 &= \frac{M}{(1 - \beta_1)((1 - a\beta_1) + (\beta_1 + a))} \tilde{M}^2 \mathbb{E} \left[ \left\| \hat{v}_0^{-1/2} \right\| \right]\end{aligned}\tag{51}$$

455 **Simple case as in [43]:** if  $\beta_1 = 0$  then  $\tilde{g}_t = g_t + m_{t+1}$  and  $g_t = \theta_t$ . Also using Lemma 3 we have  
 456 that:

$$\sum_{t=1}^{T_M} \eta_t^2 \mathbb{E} \left[ \left\| \hat{v}_t^{-1/2} g_t \right\|^2 \right] \leq \frac{\eta^2 d T_M}{(1 - \beta_2)}\tag{52}$$

457 which leads to the final bound:

$$\begin{aligned} & \mathbb{E} [\|\nabla f(w_T)\|^2] \\ & \leq \tilde{C}_1 \sqrt{\frac{d}{T_M}} + \tilde{C}_2 \frac{1}{T_M} \end{aligned} \quad (53)$$

458 where

$$\begin{aligned} \tilde{C}_1 &= C_1 + \frac{M}{(1 - a\beta_1) + (\beta_1 + a)} \left[ \frac{a(1 - \beta_1)^2}{1 - \beta_2} + 2L \frac{1}{1 - \beta_2} \right] \\ \tilde{C}_2 &= C_2 = \frac{M}{(1 - \beta_1)((1 - a\beta_1) + (\beta_1 + a))} \tilde{M}^2 \mathbb{E} [\|\hat{v}_0^{-1/2}\|] \end{aligned} \quad (54)$$

459  $\square$

## 460 E Proof of Lemma 4 (Boundedness of the iterates)

461 **Lemma.** *Given the multilayer model (8), assume the boundedness of the input data and of the loss*  
 462 *function, i.e., for any  $\xi \in \mathbb{R}^p$  and  $y \in \mathbb{R}$  there is a constant  $T > 0$  such that:*

$$\|\xi\| \leq 1 \quad \text{a.s.} \quad \text{and} \quad |\mathcal{L}'(\cdot, y)| \leq T \quad (55)$$

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** Recall that for any layer index  $\ell \in [1, L]$  we denote the output of layer  $\ell$  by  $h^{(\ell)}(w, \xi)$ :

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

463 Given the sigmoid assumption we have  $\|h^{(\ell)}(w, \xi)\| \leq 1$  for any  $\ell \in [1, L]$  and any  $(w, \xi) \in$   
 464  $\mathbb{R}^d \times \mathbb{R}^p$ . Observe that at the last layer  $L$ :

$$\begin{aligned} \|\nabla_{w^{(L)}} \mathcal{L}(\text{MLN}(w, \xi), y)\| &= \|\mathcal{L}'(\text{MLN}(w, \xi), y) \nabla_{w^{(L)}} \text{MLN}(w, \xi)\| \\ &= \|\mathcal{L}'(\text{MLN}(w, \xi), y) \sigma'(w^{(L)} h^{(L-1)}(w, \xi)) h^{(L-1)}(w, \xi)\| \\ &\leq \frac{T}{4} \end{aligned} \quad (56)$$

465 where the last equality is due to mild assumptions (55) and to the fact that the norm of the derivative  
 466 of the sigmoid function is upperbounded by 1/4.

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

$$\begin{aligned} \|w_t - \tilde{w}_{t-1}\| &= \left\| -\eta_t \hat{v}_t^{-1/2} (\theta_t + h_{t+1}) \right\| \\ &= \left\| \eta_t \hat{v}_t^{-1/2} (g_t + m_{t+1}) \right\| \\ &\leq \hat{\eta} \left\| \hat{v}_t^{-1/2} g_t \right\| + \hat{\eta} a \left\| \hat{v}_t^{-1/2} g_{t+1} \right\| \end{aligned} \quad (57)$$

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

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

468 . Thus:

$$\begin{aligned} \|w_t - \tilde{w}_{t-1}\| &\leq \hat{\eta} \left( \left\| \hat{v}_t^{-1/2} g_t \right\| + a \left\| \hat{v}_t^{-1/2} g_{t+1} \right\| \right) \\ &\leq \hat{\eta} \frac{a + 1}{\sqrt{1 - \beta_2}} \end{aligned} \quad (58)$$

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

**Proof by induction:** As in [9], 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 (56), we have

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

470 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  
 471  $\theta_{T,i}^{(L)} \geq 0$  (given the algorithm and  $\beta_1 = 0$ ) and using the fact that  $\|w_t - \tilde{w}_{t-1}\| \leq B$  we have

$$0 \leq w_{T-1,i}^{(L)} - B \leq w_{T,i}^{(L)} \leq w_{T-1,i}^{(L)} \quad (59)$$

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 (59) 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

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

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

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

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

$$\nabla_{w^{(\ell)}} \mathcal{L}(\text{MLN}(w, \xi), y) = \mathcal{L}'(\text{MLN}(w, \xi), y) \left( \prod_{j=1}^{\ell+1} \sigma' \left( w^{(j)} h^{(j-1)}(w, \xi) \right) \right) h^{(\ell-1)}(w, \xi) \quad (60)$$

This last quantity is bounded as long as we can prove that for any layer  $\ell$  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 (60) 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  $\ell$  satisfy

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

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



## F Comparison to some related methods

**Comparison to nonconvex optimization works.** Recently, [40, 5, 38, 42, 44, 22] provide some theoretical analysis of ADAM-type algorithms when applying them to smooth nonconvex optimization problems. For example, [5] provides a bound, which is  $\min_{t \in [T]} \mathbb{E}[\|\nabla f(w_t)\|^2] = \mathcal{O}(\log T / \sqrt{T})$ . Yet, this data independent bound does not show any advantage over standard stochastic gradient descent. Similar concerns appear in other papers.

To get some adaptive data dependent bound that are in terms of the gradient norms observed along the trajectory) when applying OPT-AMSGRAD to nonconvex optimization, one can follow the approach of [2] or [6]. They provide ways to convert algorithms with adaptive data dependent regret bound for convex loss functions (e.g. ADAGRAD) to the ones that can find an approximate stationary point of nonconvex loss functions. Their approaches are modular so that simply using OPT-AMSGRAD as the base algorithm in their methods will immediately lead to a variant of OPT-AMSGRAD that enjoys some guarantee on nonconvex optimization. The variant can outperform the ones instantiated by other ADAM-type algorithms when the gradient prediction  $m_t$  is close to  $g_t$ . The details are omitted since this is a straightforward application.

**Comparison to AO-FTRL [26].** In [26], the authors propose AO-FTRL, which has the update of the form  $w_{t+1} = \arg \min_{w \in \Theta} (\sum_{s=1}^t g_s)^\top w + m_{t+1}^\top w + r_{0:t}(w)$ , where  $r_{0:t}(\cdot)$  is a 1-strongly convex loss function with respect to some norm  $\|\cdot\|_{(t)}$  that may be different for different iteration  $t$ . Data dependent regret bound was provided in the paper, which is  $r_{0:T}(w^*) + \sum_{t=1}^T \|g_t - m_t\|_{(t)}^*$  for any benchmark  $w^* \in \Theta$ . We see that if one selects  $r_{0:t}(w) := \langle w, \text{diag}\{\hat{v}_t\}^{1/2} w \rangle$  and  $\|\cdot\|_{(t)} := \sqrt{\langle \cdot, \text{diag}\{\hat{v}_t\}^{1/2} \cdot \rangle}$ , then the update might be viewed as an optimistic variant of ADAGRAD. However, no experiments was provided in [26].

**Comparison to OPTIMISTIC-ADAM [8].** We are aware that [8] proposed one version of optimistic algorithm for ADAM, which is called OPTIMISTIC-ADAM in their paper. A slightly modified version is summarized in Algorithm 4. Here, OPTIMISTIC-ADAM+ $\hat{v}_t$  is OPTIMISTIC-ADAM in [8] with the additional max operation  $\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$  to guarantee that the weighted second moment is monotone increasing.

---

### Algorithm 4 OPTIMISTIC-ADAM [8]+ $\hat{v}_t$ .

---

- 1: Required: parameter  $\beta_1, \beta_2$ , and  $\eta_t$ .
  - 2: Init:  $w_1 \in \Theta$  and  $\hat{v}_0 = v_0 = \epsilon 1 \in \mathbb{R}^d$ .
  - 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**
- 

We want to emphasize that the motivations are different. OPTIMISTIC-ADAM in their paper is designed to optimize two-player games (e.g. GANs [15]), while the proposed algorithm in this paper is designed to accelerate optimization (e.g. solving empirical risk minimization quickly). [8] focuses on training GANs [15]. GANs is a two-player zero-sum game. There have been some related works in OPTIMISTIC ONLINE LEARNING like [7, 30, 34]) showing that if both players use some kinds of OPTIMISTIC-update, then accelerating the convergence to the equilibrium of the game is possible. [8] was inspired by these related works and showed that OPTIMISTIC-MIRROR-DESCENT can avoid the cycle behavior in a bilinear zero-sum game, which accelerates the convergence. Furthermore, [8] did not provide theoretical analysis of OPTIMISTIC-ADAM.

## 515 G Additional Remarks and Runs on the Gradient Prediction Process

516 **Two illustrative examples.** We provide two toy examples to demonstrate how OPT-AMSGRAD  
 517 works with the chosen extrapolation method. First, consider minimizing a quadratic function  
 518  $H(w) := \frac{b}{2}w^2$  with vanilla gradient descent method  $w_{t+1} = w_t - \eta_t \nabla H(w_t)$ . The gradient  
 519  $g_t := \nabla H(w_t)$  has a recursive description as  $g_{t+1} = bw_{t+1} = b(w_t - \eta_t g_t) = g_t - b\eta_t g_t$ . So,  
 520 the update can be written in the form of  $g_t = Ag_{t-1} + \mathcal{O}(\|g_{t-1}\|_2^2)u_{t-1}$ , with  $A = (1 - b\eta)$  and  
 521  $u_{t-1} = 0$  by setting  $\eta_t = \eta$  (constant step size). Therefore, the extrapolation method should predict  
 522 well.

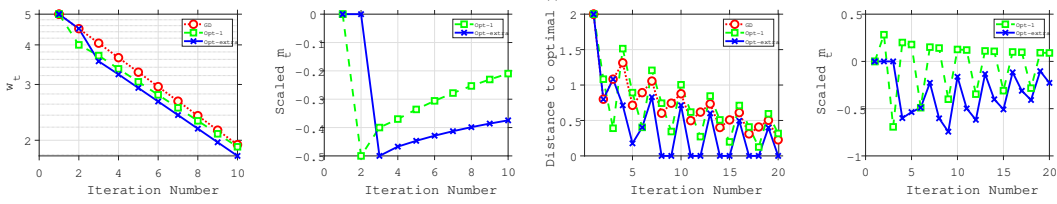


Figure 4: (a): The iterate  $w_t$ ; the closer to the optimal point 0 the better. (b): A scaled and clipped version of  $m_t$ :  $w_t - w_{t-1/2}$ , which measures how the prediction of  $m_t$  drives the update towards the optimal point. In this scenario, the more negative the better. (c): Distance 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.

523 Specifically, consider optimizing  $H(w) := w^2/2$  by the following three algorithms with the same  
 524 step size. One is Gradient Descent (GD):  $w_{t+1} = w_t - \eta_t g_t$ , while the other two are OPT-  
 525 AMSGRAD with  $\beta_1 = 0$  and the second moment term  $\hat{v}_t$  being dropped:  $w_{t+\frac{1}{2}} = \Pi_{\Theta}[w_{t-\frac{1}{2}} - \eta_t g_t]$ ,  
 526  $w_{t+1} = \Pi_{\Theta}[w_{t+\frac{1}{2}} - \eta_{t+1} m_{t+1}]$ . We denote the algorithm that sets  $m_{t+1} = g_t$  as Opt-1, and denote  
 527 the algorithm that uses the extrapolation method to get  $m_{t+1}$  as Opt-extra. We let  $\eta_t = 0.1$  and the  
 528 initial point  $w_0 = 5$  for all the three methods. The simulation results are on Figure 4 (a) and (b).  
 529 Sub-figure (a) plots update  $w_t$  over iteration, where the updates should go towards the optimal point  
 530 0. Sub-figure (b) is about a scaled and clipped version of  $m_t$ , defined as  $w_t - w_{t-1/2}$ , which can be  
 531 viewed as  $-\eta_t m_t$  if the projection (if exists) is lifted. Sub-figure (a) shows that Opt-extra converges  
 532 faster than the other methods. Furthermore, sub-figure (b) shows that the prediction by the extrap-  
 533 olation method is better than the prediction by simply using the previous gradient. The sub-figure  
 534 shows that  $-m_t$  from both methods all point to 0 in all iterations and the magnitude is larger for the  
 535 one produced by the extrapolation method after iteration 2.<sup>2</sup>

536 Now let us consider another problem: an online learning problem proposed in [31]<sup>3</sup>. Assume the  
 537 learner's decision space is  $\Theta = [-1, 1]$ , and the loss function is  $\ell_t(w) = 3w$  if  $t \bmod 3 = 1$ , and  
 538  $\ell_t(w) = -w$  otherwise. The optimal point to minimize the cumulative loss is  $w^* = -1$ . We  
 539 let  $\eta_t = 0.1/\sqrt{t}$  and the initial point  $w_0 = 1$  for all the three methods. The parameter  $\lambda$  of the  
 540 extrapolation method is set to  $\lambda = 10^{-3} > 0$ . The results are on Figure 4 (c) and (d). Sub-figure  
 541 (c) shows that Opt-extra converges faster than the other methods while Opt-1 is not better than GD.  
 542 The reason is that the gradient changes from  $-1$  to  $3$  at  $t \bmod 3 = 1$  and it changes from  $3$  to  $-1$   
 543 at  $t \bmod 3 = 2$ . Consequently, using the current gradient as the guess for the next clearly is not a  
 544 good choice, since the next gradient is in the opposite direction of the current one. Sub-figure (d)  
 545 shows that  $-m_t$  by the extrapolation method always points to  $w^* = -1$ , while the one by using  
 546 the previous negative direction points to the opposite direction in two thirds of rounds. It shows  
 547 that the extrapolation method is much less affected by the gradient oscillation and always makes the  
 548 prediction in the right direction, which suggests that the method can capture the aggregate effect.

### 549 G.1 Choice of parameter $r$

550 Since the number of past gradients  $r$  is important in our algorithm, we compare Figure 5 the perfor-  
 551 mance under different values  $r = 3, 5, 10$  on two datasets. From the result we see that the choice  
 552 of  $r$  does not have significant impact on the training loss. Taking into consideration both quality of

<sup>2</sup> The extrapolation method needs at least two gradients for prediction. This is why in the first two iterations,  $m_t$  is 0.

<sup>3</sup> [31] uses this example to show that ADAM [19] fails to converge.

553 gradient prediction and computational cost,  $r = 5$  is a good choice for most applications here. We  
 554 remark that empirically, the performance comparison among  $r = 3, 5, 10$  is not absolutely consistent  
 555 (i.e. more means better) in all cases. One possible reason is that for deep neural networks, the high  
 556 diversity of gradients computed through the iterations, due to the nonconvexity of the loss, makes  
 557 most of them inefficient for the predictable process  $\{m_t\}_{t>0}$ . Only recent ones ( $r \leq 5$ ) are useful.

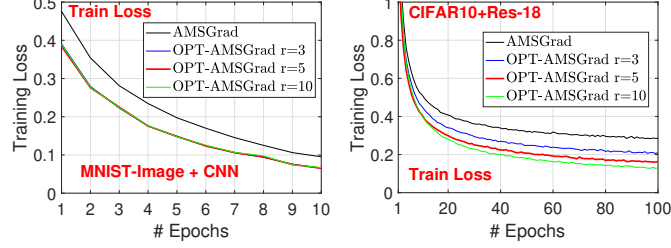


Figure 5: Training loss w.r.t.  $r$ .

## 558 H Additional Numerical Experiments

559 Additional experiments are presented Figure 6 to highlight the acceleration effect of OPT-  
 560 AMSGRAD at early stage.

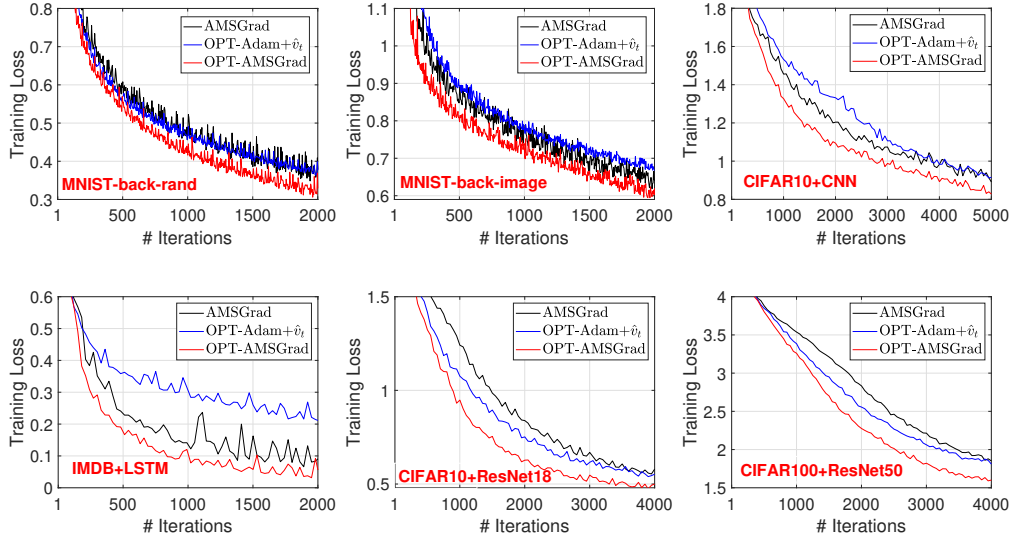


Figure 6: Training loss vs. Number of iterations for fully connected NN, LSTM, CNN and ResNet.