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

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper, we propose a new variant of AMSGrad [31], a popular adaptive gradient based optimization algorithm widely used in training deep neural networks. Our algorithm adds prior knowledge about the sequence of consecutive mini-batch gradients leveraging an underlying structure which makes 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 versions of nonconvex optimization. In the nonconvex case, we establish a $\mathcal{O}\left(\sqrt{d/T} + d/T\right)$ non-asymptotic bound independent of the initialization of the method. We illustrate the practical speedup on several deep learning models through numerical experiments.

1 Introduction

Deep learning models have been successful in several applications, from robotics (e.g. [20]), computer vision (e.g [17, 14]), reinforcement learning (e.g. [25]), to natural language processing (e.g. [15]). With the sheer size of modern data sets 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 [31], ADAM [18], RMSPROP [35], ADADELTA [41], and NADAM [10].

All the prevalent algorithms for training deep networks mentioned above combine two ideas: the idea of adaptivity from ADAGRAD [11, 23] and the idea of momentum from NESTEROV'S METHOD [27] or HEAVY BALL method [28]. 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 gradient in each dimension which helps in exploiting the geometry of data. On the other hand, NESTEROV'S METHOD or HEAVY BALL Method [28] is an accelerated optimization algorithm whose update not only depends on the current iterate and current gradient but also depends on the past gradients (i.e. momentum). State-of-the-art algorithms like AMSGRAD [31] and ADAM [18] leverage these ideas to accelerate the training process of highly nonconvex objective functions such as deep neural networks losses.

In this paper, we propose an algorithm that goes further than the hybrid of the adaptivity and momentum approach. Our algorithm is inspired by OPTIMISTIC ONLINE LEARNING [7, 29, 34, 1, 24], which assumes that a good *predictable process* of the gradient of the loss function in each round of online learning is available, and plays an action by exploiting these predictors. By exploiting this (possibly) arbitrary process, algorithms in OPTIMISTIC ONLINE LEARNING enjoy smaller regret than the ones without. We combine the OPTIMISTIC ONLINE LEARNING idea with 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 in their paper leveraging optimistic online mirror descent [30]. Yet, OPTIMISTIC-ADAM is specifically designed to optimize two-player games (e.g. GANs [14]). 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.

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 good 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 both an underlying scheme which makes the gradients sequentially predictable and the global parameter learning.
- We focus on the *theoretical* justifications of our method by establishing novel *non-asymptotic* and *global* convergence rates in both the convex and nonconvex case. Based on both *convex regret minimization* and *nonconvex stochastic optimization* views, we prove, respectively, that our algorithm suffers regret of $\mathcal{O}(\sqrt{\sum_{t=1}^{T} \|g_t - m_t\|^2_{\psi_{t-1}}})$ 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 convergence of loss function, 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 called 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 problem 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 [18, 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 T vectors $g_1, g_2, \dots, g_T \in \mathbb{R}^d$.

2 Preliminaries

Optimistic Online learning. The standard setup of ONLINE LEARNING is that, in each round t, an online learner selects an action $w_t \in \Theta \subseteq \mathbb{R}^d$, then the learner observes $\ell_t(\cdot)$ and suffers 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[1]. Consider the FOLLOW-THE-REGULARIZED-LEADER (FTRL, [16]) online learning algorithm which update reads

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

where η is a parameter, $\mathsf{R}(\cdot)$ is a 1-strongly convex function with respect to a norm ($\|\cdot\|$) 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 $O(\sqrt{\sum_{t=1}^{T} \|g_t\|_*^2})$. The update of its optimistic variant, noted OPTIMISTIC-FTRL and developed in [34] reads

$$w_t = \arg\min_{w \in \Theta} \langle w, L_{t-1} + m_t \rangle + \frac{1}{\eta} \mathsf{R}(w) \ , \tag{2}$$

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 $O(\sqrt{\sum_{t=1}^{T} \|g_t - m_t\|_*^2})$.

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 bounds up to a factor 2. Yet, when the predictions are well constructed, 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 present Section 5 a simple yet effective predictable process $\{m_t\}_{t>0}$ leading to an empirical evidence of acceleration.

Adaptive optimization methods. Recently, adaptive optimization has been popular in various deep learning applications due to their superior empirical performance. ADAM [18] is a very popular adaptive algorithm for training deep neural networks. It combines the momentum idea [28] with the idea of ADAGRAD [11], which has different learning rates for different dimensions, adaptive to the learning process. More specifically, the learning rate of ADAGRAD in iteration t for dimension j is proportional to the inverse of $\sqrt{\Sigma_{s=1}^{t} g_s[j]^2}$, 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 networks, it is observed that the learning rate might decay too fast [18]. Therefore, [18] 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 at some online convex optimization problems. 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, $\frac{\eta_t}{\sqrt{\hat{v}_t}}$, which helps for the convergence (i.e. average regret $\mathcal{R}_T/T \to 0$).

Algorithm 1 AMSGRAD [31]

1: Required: parameter β_1, β_2, and η_t.
2: Init: $w_1 \in \Theta \subseteq \mathbb{R}^d$ 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: $w_{t+1} = w_t - \eta_t \frac{\theta_t}{\sqrt{\hat{v}_t}}$. (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 also updates an auxiliary variable $\tilde{w}_{t+1} \in \Theta$ (line 8) and finally updates the current model parameter (line 9).

[1]Imagine that if the learner would had been 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 minimizes the regret.

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 $h_{t+1} := \beta_1 \theta_{t-1} + (1 - \beta_1) m_{t+1}$ and $m_{t+1}$
       is the guess of $g_{t+1}$.
10: end for
```

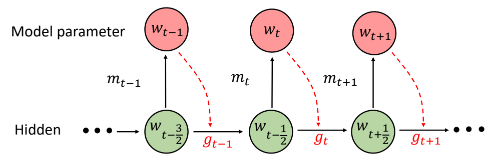

Figure 1: OPT-AMSGRAD UNDERLYING STRUCTURE.

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 and get a single update as $w_{t+1} = \tilde{w}_t - \eta_t \frac{\theta_t}{\sqrt{\hat{v}_t}} - \eta_t \frac{h_{t+1}}{\sqrt{\hat{v}_t}}$. Compared to AMSGRAD, the algorithm is characterized by a two-level update that interlink 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} of g_{t+1}, see Figure 1 for a schematic illustration. 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 procedure plays naturally an important role and will be tackled Section 5.

The proposed OPT-AMSGRAD inherits three properties:

- Adaptive learning rate of each dimension as ADAGRAD [11]. (line 6, line 8 and line 9)
- Exponential moving average of the past gradients as NESTEROV'S METHOD [27] and the HEAVY-BALL method [28]. (line 5)
- Optimistic update that exploits a good guess of the next gradient vector as optimistic online learning algorithms [7, 29, 34]. (line 9)

The first property helps for acceleration when the gradient has a sparse structure. The second one is from the well-recognized idea of momentum which can also help for acceleration. The last one, perhaps less known outside the ONLINE LEARNING community, can actually lead to acceleration when the prediction of the next gradient is good. This property will be elaborated in the following subsection in which we provide the theoretical analysis of OPT-AMSGRAD. Observe that the proposed algorithm does not reduce to AMSGRAD when $m_t = 0$.

4 Global Convergence of OPT-AMSGRAD

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

Notations. To begin with, let us introduce some notations first. We denote the Mahalanobis norm $\| \cdot \|_H := \sqrt{\langle \cdot, H \cdot \rangle}$ for some 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 whose i_{th} diagonal element is $\hat{v}_t[i]$ in Algorithm 2. We define its corresponding Mahalanobis norm $\| \cdot \|_{\psi_t} := \sqrt{\langle \cdot, \text{diag}\{\hat{v}_t\}^{1/2} \cdot \rangle}$, where we abuse the notation ψ_t to represent the PSD matrix $H_t^{1/2} := \text{diag}\{\hat{v}_t\}^{1/2}$. Consequently, $\psi_t(\cdot)$ is 1-strongly convex with respect to the norm $\| \cdot \|_{\psi_t} := \sqrt{\langle \cdot, \text{diag}\{\hat{v}_t\}^{1/2} \cdot \rangle}$. Namely, $\psi_t(\cdot)$ 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. A consequence of 1-strongly 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 $B_{\psi_t}(u, v)$ 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 generating function. We can also define the corresponding dual norm $\| \cdot \|_{\psi_t^*} := \sqrt{\langle \cdot, \text{diag}\{\hat{v}_t\}^{-1/2} \cdot \rangle}$.

4.1 Convex Regret Analysis

We prove the following result regarding the regret in the convex optimization setting. That is, we assume that the loss functions $\{\ell_t\}_{t>0}$ are convex. We also assume that Θ has bounded diameter D_∞, which is a standard assumption in previous works [31, 18] on adaptive methods. It is necessary in regret analysis since if the boundedness assumption is lifted, one might construct a scenario such that the benchmark is $w^* = \infty$ and the learner's regret is infinite.

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

$$\mathcal{R}_T \leq \frac{B_{\psi_1}(w^*, \tilde{w}_1)}{\eta_1} + \sum_{t=1}^{T} \frac{\eta_t}{2} \|g_t - \tilde{m}_t\|^2_{\psi^*_{t-1}} + D^2_\infty \sum_{t=1}^{T} \left[\beta_1^2 \|g_t - \theta_{t-1}\|_{\psi^*_{t-1}} + \frac{1}{\eta_{\min}} \hat{v}_T^{1/2}[i] \right] , \tag{3}$$

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

Corollary 1. *Suppose $\beta_1 = 0$ and $\{v_t\}_{t>0}$ is an increasing monotone sequence, then we obtain the following regret bound for any $w^* \in \Theta$ and sequence $\{\eta_t\}_{t>0}$:*

$$\mathcal{R}_T \leq \frac{B_{\psi_1}(w^*, \tilde{w}_1)}{\eta_1} + \sum_{t=1}^{T} \frac{\eta_t}{2} \|g_t - m_t\|^2_{\psi^*_{t-1}} + \frac{D^2_\infty}{\eta_{\min}} \sum_{i=1}^{d} \left[(1-\beta_2) \sum_{s=1}^{T} \beta_2^{T-s} (g_s[i] - m_s[i])^2 \right]^{1/2} , \tag{4}$$

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

For convex regret minimization, the resutls above yields that the learner suffers regret of $\mathcal{O}(\sqrt{\sum_{t=1}^{T} \|g_t - m_t\|^2_{\psi_{t-1}}})$ with an access to an arbitrary predictable process $\{m_t\}_{t>0}$ of the mini-batch gradient. The better the predictors, the lower the regret will be. One can thus wonder how the learner can build those good gradients predictions $\{m_t\}_{t>0}$. Is this process can be learnt through the iterations?

Those questions are interesting research questions and will not be dealt in this paper for the sake of page limit. Though, for implementation purposes, we derive a simple, yet effective, gradient prediction algorithm, see Algorithm 3 in Section 5 to embed to our new OPT-AMSGRAD method.

4.2 Nonconvex Analysis (Finite-Time Upper Bound)

In this section, we discuss the offline and stochastic non-convex optimization properties of our online framework. In the stochastic optimization literature, the problem we are tackling reads as follows:

$$\min_{w \in \Theta} f(w) := \mathbb{E}[f(w, \xi)] , \tag{5}$$

where ξ is some random noise and only noisy versions of the objective function are accessible in this work. The objective function $f(w)$ is (potentially) nonconvex and has Lipschitz gradients.

Set the terminating iteration number, $T \in \{0, \ldots, T_{\mathsf{max}} - 1\}$, as a discrete r.v. with:

$$P(T = \ell) = \frac{\eta_\ell}{\sum_{j=0}^{T_{\mathsf{max}}-1} \eta_j} , \tag{6}$$

where T_{max} is the maximum number of iteration. The random termination number (6) is inspired by [13] which enables one to show non-asymptotic convergence to stationary point for non-convex optimization.

We make the following mild assumptions necessary to our analysis:

H1. *For any $t > 0$, the estimated weight w_t stays within a ℓ_∞−ball. There exists a constant $W > 0$ such that $\|w_t\| \leq W$ almost surely.*

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

$$f(w) - f(\vartheta) - \nabla f(\vartheta)^\top (w - \vartheta) \leq \frac{L}{2} \|w - \vartheta\|^2 .$$

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

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

Classically in nonconvex optimization [13] we make an assumption on the magnitude of the gradient:

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

We begin with some auxiliary Lemmas important for the analysis. The first one ensures bounded norms of various quantities of interests (resulting from the classical stochastic gradient boundedness assumption):

Lemma 1. *Assume assumption H4, then the quantities defined in Algorithm 2 satisfy for any $w \in \Theta$ and $t > 0$:*

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

Then, following [39] and their study of the SGD with Momentum (not AMSGrad but simple momentum) we denote for any $t > 0$:

$$\overline{w}_t = w_t + \frac{\beta_1}{1-\beta_1}(w_t - \tilde{w}_{t-1}) = \frac{1}{1-\beta_1} w_t - \frac{\beta_1}{1-\beta_1} \tilde{w}_{t-1} \; , \tag{7}$$

and derive an important Lemma:

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

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

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

Lemma 3. *Assume H4, a strictly positive and a sequence of constant stepsizes $\{\eta_t\}_{t>0}$, $\beta \in [0,1]$, then the following holds:*

$$\sum_{k=1}^{T_{\mathsf{max}}} \eta_t^2 \mathbb{E}\left[\left\| \hat{v}_t^{-1/2} \theta_t \right\|_2^2 \right] \leq \frac{\eta^2 d T_{\mathsf{max}} (1-\beta_1)}{(1-\beta_2)(1-\gamma)} \; .$$

We now formulate the main result of our paper giving a finite-time upper bound of the quantity $\mathbb{E}\left[\|\nabla f(w_T)\|^2\right]$ where T is a random termination number distributed according to 6, see [13].

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:*

$$\mathbb{E}\left[\|\nabla f(w_T)\|^2\right] \leq \tilde{C}_1 \sqrt{\frac{d}{T_{\mathsf{max}}}} + \tilde{C}_2 \frac{1}{T_{\mathsf{max}}} \; , \tag{8}$$

where K is a random termination number distributed according (6).

We remark that the bound for our OPT-AMSGrad method matches the complexity bound of $\mathcal{O}\left(\sqrt{d/T_{\mathsf{max}}} + 1/T_{\mathsf{max}}\right)$ of [13] for SGD and [43] for AMSGrad method.

4.3 Checking H1 for a Deep Neural Network

We show in this section that the weights satisfy assumption H1 and stay in a bounded set when the model we are fitting, using our method, is a fully connected feed forward neural network. The activation function for this section will be sigmoid function and we add a ℓ_2 regularization.

For the sake of notation, we assume $\beta_1 = 0$. We consider a fully connected feed forward neural network with L layers modeled by the function $\mathsf{MLN}(w, \xi) : \mathbb{R}^l \to \mathbb{R}$:

$$\mathsf{MLN}(w, \xi) = \sigma\left(w^{(L)} \sigma\left(w^{(L-1)} \ldots \sigma\left(w^{(1)} \xi \right) \right) \right) \tag{9}$$

where $w = [w^{(1)}, w^{(2)}, \cdots, w^{(L)}]$ is the vector of parameters, $\xi \in \mathbb{R}^l$ is the input data and σ is the sigmoid activation function. We assume a l dimension input data and a scalar output for simplicity. The stochastic objective function (5) reads:

$$f(w, \xi) = \mathcal{L}(\mathsf{MLN}(w, \xi), y) + \frac{\lambda}{2} \|w\|^2 \tag{10}$$

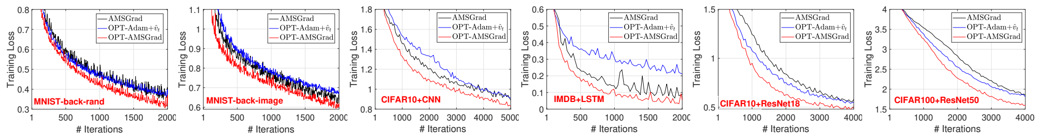

Figure 2: Training loss vs. Number of iterations. The first row are results with fully-connected NN.

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 layer index $\ell \in [1, L]$ we denote the output of layer ℓ by $h^{(\ell)}(w, \xi) = \sigma\left(w^{(\ell)}\sigma\left(w^{(\ell-1)}\ldots\sigma\left(w^{(1)}\xi\right)\right)\right)$.

The following Lemma proves that assumption H1 is satisfied with a feed forward neural net (9):

Lemma 4. *Given the multilayer model* (9)*, assume the boundedness of the input data and of the loss function,* i.e., *for any $\xi \in \mathbb{R}^l$ and $y \in \mathbb{R}$ there is a constant $T > 0$ such that $\|\xi\| \leq 1$ a.s. and $|\mathcal{L}'(\cdot, y)| \leq T$ 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 $\left\|w^{(\ell)}\right\| \leq A_{(\ell)}$*

5 Numerical Experiments

5.1 Gradient Estimation

From the analysis in the previous section, we know that whether OPT-AMSGRAD converges faster than its counterpart depends on how m_t is chosen. In OPTIMISTIC-ONLINE LEARNING, m_t is usually set to $m_t = g_{t-1}$, which means that it uses the previous gradient as a guess of the next one. The choice can accelerate the convergence to equilibrium in some two-player zero-sum games [29, 34, 8], in which each player uses an optimistic online learning algorithm against its opponent.

However, this paper is about solving optimization problems, as in (5), instead of solving zero-sum games. In most classical deep learning tasks, as we will develop in the numerical section, (5) even reads $\min_{w \in \Theta} f(w) = \sum_{i=1}^{n} f(w, \xi_i)$ for a fixed batch of n samples $\{\xi_i\}_{i=1}^{n}$. We propose to use the extrapolation algorithm of [32]. Extrapolation studies estimating the limit of sequence using the last few iterates [3]. Some classical works include Anderson acceleration [37], minimal polynomial extrapolation [4], reduced rank extrapolation [12]. These methods typically assume that the sequence $\{x_t\} \in \mathbb{R}^d$ has a linear relation $x_t = A(x_{t-1} - x^*) + x^*$ and $A \in \mathbb{R}^{d \times d}$ is an unknown, not necessarily symmetric, matrix. The goal is to find the fixed point of x^*. [32] relaxes the assumption to certain degrees. It assumes that the sequence $\{x_t\} \in \mathbb{R}^d$ satisfies

$$x_t - x^* = A(x_{t-1} - x^*) + e_t, \tag{11}$$

where e_t is a second order term satisfying $\|e_t\|_2 = O(\|x_{t-1} - x^*\|_2^2)$ and $A \in \mathbb{R}^{d \times d}$ is an unknown matrix. The extrapolation algorithm we used is shown in Algorithm 3. Some theoretical guarantees regarding the distance between the output and x^* are provided in [32]. For our numerical experi-

Algorithm 3 REGULARIZED APPROXIMATE MINIMAL POLYNOMIAL EXTRAPOLATION (RMPE) [32]

1: **Input:** sequence $\{x_s \in \mathbb{R}^d\}_{s=0}^{s=r}$, parameter $\lambda > 0$.
2: Compute matrix $U = [x_1 - x_0, \ldots, x_r - x_{r-1}] \in \mathbb{R}^{d \times r}$.
3: Obtain z by solving $(U^\top U + \lambda I)z = \mathbf{1}$.
4: Get $c = z/(z^\top \mathbf{1})$.
5: **Output:** $\Sigma_{i=0}^{r-1} c_i x_i$, the approximation of the fixed point x^*.

ments in the next section, we run OPT-AMSGRAD using Algorithm 3 to get m_t. Specifically, m_t is obtained by (a) calling Algorithm 3 with input being a sequence of some past $r+1$ gradients, $\{g_t, g_{t-1}, g_{t-2}, \ldots, g_{t-r}\}$, where r is a parameter and (b) setting $m_t := \Sigma_{i=0}^{r-1} c_i g_{t-r+i}$ from the output of 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

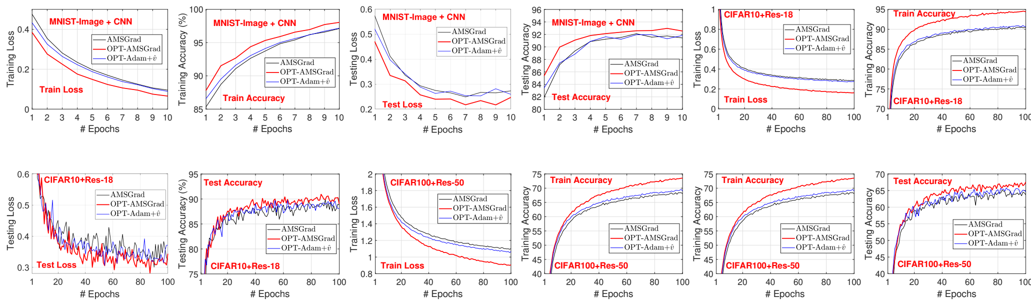

Figure 3: *MNIST-back-image* + CNN, *CIFAR10* + Res-18 and *CIFAR100* + Res-50 . We compare three methods in terms of training (cross-entropy) loss and accuracy, testing loss and accuracy.

underlying function f). Then, we might rewrite (11) as

$$g_t = A g_{t-1} + O(\|g_{t-1}\|_2^2) u_{t-1}, \tag{12}$$

for some vector u_{t-1} with a unit norm. The 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 whose length is $O(\|g_{t-1}\|_2^2)$. If the algorithm is guaranteed to converge to a stationary point, the magnitude of the error component will eventually go to zero. We remark that the choice of algorithm for gradient prediction is surely not unique. We propose to use the recent result among various related works. Indeed, one can use any method that can provide reasonable guess of gradient in next iteration.

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 OPTIMISTIC-AMSGRAD.

Methods. We consider two baselines. The first one is the original AMSGRAD. The hyper-parameters are set to be β_1 and β_2 to be 0.9 and 0.999 respectively, as recommended by [31]. We tune the learning rate η over a fine grid and report the best result. The other competing method is the aforementioned OPTIMISTIC-ADAM+$\hat{v}_t$ method, see [8]. The key difference is that it uses previous gradient as the gradient prediction of the next iteration. We also report the best result achieved by tuning the step size η for OPTIMISTIC-ADAM+$\hat{v}_t$. For OPTIMISTIC-AMSGRAD, we use the same β_1, β_2 and the best step size η of AMSGRAD for a fair evaluation of the improvement brought by the extra optimistic step. Yet, OPTIMISTIC-AMSGRAD has an additional parameter r that controls the number of previous gradients used for gradient prediction. Fortunately, we observe similar performance of OPTIMISTIC-AMSGRAD with different values of r. Hence, we report $r = 5$ for now when comparing with other baselines. We will address on the choice of r at the end of this section. In all experiments, all the optimization algorithms are initialized at the same point. We report the results averaged over 5 repetitions.

Datasets. Following [31] and [18], we compare different algorithms on *MNIST*, *CIFAR10*, *CIFAR100*, and *IMDB* datasets. For *MNIST*, we use two noisy variants named as 1.65*MNIST-back-rand* and 1.65*MNIST-back-image* from [19]. They both have 12000 training samples and 50000 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, whose pixel values generated uniformly from 0 to 255, while *MNIST-back-image* takes random patches from a black and white as noisy background. The input dimension is 784 (28×28) and the number of classes is 10. *CIFAR10* and *CIFAR100* are popular computer-vision datasets consisting of 50000 training images and 10000 test images, of size 32×32. The number of classes are 10 and 100, respectively. The *IMDB* movie review dataset is a binary classification dataset with 25000 training and testing samples respectively. It is a popular datasets for text classification.

Network architecture. We adopt a multi-layer fully-connected neural network with input layer followed by a hidden layer with 200 nodes, which is connected to another layer with 100 nodes before the output layer. The activation function is ReLU for hidden layers, and softmax for the output layer. This network is tested on *MNIST* variants. Since convolutional networks are popular for image classification tasks, we consider an ALL-CNN architecture proposed by [33], which is constructed with several convolutional blocks and dropout layers. In addition, we also apply residual networks, Resnet-18 and Resnet-50 [17], which have achieved many state-of-the-art results. For the texture *IMDB* dataset, we consider training a Long-Short Term Memory (LSTM) network. The network includes a word embedding layer with 5000 input entries representing most frequent words in the dataset, and each word is embedded into a 32 dimensional space. The output of the embedding layer is passed to 100 LSTM units, which is then connected to 100 fully connected ReLu's before the output layer. For all the models, we use cross-entropy loss. A mini-batch size of 128 is used to compute the stochastic gradients.

Results. Firstly, to illustrate the acceleration effect of Optimistic-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 Optimistic-AMSGrad converges faster than the other competing methods, right after the training begins. In other words, we need fewer iterations (samples) to achieve the same training loss. This validates one of the main advantages of Optimistic-AMSGrad, which is a higher sample efficiency. We are also curious about the long-term performance and generalization of the proposed method in test phase. In Figure 3, we plot the corresponding results when the model is trained to the state with stable test accuracy. We observe: 1) In the long term, Optimistic-AMSGrad algorithm may converge to a better point with smaller objective function value, and 2) In this three applications, the proposed Optimistic-AMSGrad also outperforms the competing methods in terms of test accuracy. These are also important benefits of Optimistic-AMSGrad.

5.3 Choice of parameter r

Recall that our proposed algorithm has the parameter r that governs the use of past information. Figure 4 compares the performance under different values of $r = 3, 5, 10$ on two datasets. From the result we see that the choice of r does not have significant impact on learning performance. Taking into consideration both quality of gradient prediction and computational cost, it appears that $r = 5$ is a good choice. We remark that empirically, the performance comparison among $r = 3, 5, 10$ is not absolutely consistent (i.e. more means better) in all cases. One possible reason is that for deep neural nets (with highly non-convex loss), using gradient information from too long ago may not be helpful in accurate gradient prediction. Nevertheless, $r = 5$ seems to be good for most applications.

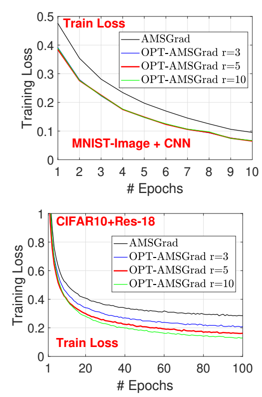

Figure 4: Training loss with different r.

5.4 Some Remarks on the Experiments

Discussion on the iteration cost: We observe that the iteration cost (i.e., actual running time per iteration) of our implementation of Optimistic-AMSGrad with $r = 5$ is roughly two times larger than the standard AMSGrad. When $r = 3$, the cost is roughly 0.7 times longer. Nevertheless, Optimistic-AMSGrad may still be beneficial in terms of training efficiency, since fewer iterations are typically needed. For example, in Figure 3, to reach the training loss of AMSGrad at 100 epochs, the proposed method only needs roughly 20 and 40 epochs, respectively. That said, Optimistic-AMSGrad needs 40% and 80% time to achieve same training loss as AMSGrad, in this two problems.

The computational overhead mostly comes from the gradient extrapolation step. More specifically, recall that the extrapolation step consists of: (a) The step of constructing the linear system $(U^\top U)$. The cost of this step can be optimized and reduced to $\mathcal{O}(d)$, since the matrix U only changes one column at a time. (b) The step of solving the linear system. The cost of this step is $O(r^3)$, which is negligible as the linear system is very small (5-by-5 if $r = 5$). (c) The step that outputs an estimated gradient as a weighted average of previous gradients. The cost of this step is $\mathcal{O}(r \times d)$. Thus, the computational overhead is $\mathcal{O}\left((r+1)d + r^3\right)$. Yet, we notice that step (a) and (c) is parallelizable, so they can be accelerated in practice.

Memory usage: Our algorithm needs a storage of past r gradients for each coordinate, in addition to the estimated second moments and the moving average. Though it seems demanding compared to the standard AMSGrad, it is relatively cheap compared to Natural gradient method (e.g., [22]), as Natural gradient method needs to store some matrix inverse.

6 Conclusion

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

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

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *NIPS*, 2014.

[15] A. Graves, A. rahman Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. *ICASSP*, 2013.

[16] E. Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2016.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, 2016.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[19] 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.

[20] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *NIPS*, 2017.

[21] X. Li and F. Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. *AISTAT*, 2019.

[22] J. Martens and R. Grosse. Optimizing neural networks with kronecker-factored approximate curvature. *ICML*, 2015.

[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,* OPTIMISTIC-AMSGRAD *(Algorithm 2) has regret*

$$\begin{aligned}\mathcal{R}_T \leq & \frac{1}{\eta_{\min}} D_\infty^2 \sum_{i=1}^{d} \hat{v}_T^{1/2}[i] + \frac{B_{\psi_1}(w^*, \tilde{w}_1)}{\eta_1} + \sum_{t=1}^{T} \frac{\eta_t}{2} \|g_t - \tilde{m}_t\|^2_{\psi^*_{t-1}} \\ & + D_\infty^2 \beta_1^2 \sum_{t=1}^{T} \|g_t - \theta_{t-1}\|_{\psi^*_{t-1}} .\end{aligned} \tag{13}$$

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

Proof Beforehand, note:

$$\begin{aligned}\tilde{g}_t &= \beta_1 \theta_{t-1} + (1-\beta_1) g_t \\ \tilde{m}_{t+1} &= \beta_1 \theta_{t-1} + (1-\beta_1) m_{t+1}\end{aligned} \tag{14}$$

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

$$\begin{aligned}& \textit{Regret}_T := \sum_{t=1}^{T} \ell_t(w_t) - \min_{w \in \Theta} \sum_{t=1}^{T} \ell_t(w) \\ \leq & \sum_{t=1}^{T} \langle w_t - w^*, \nabla \ell_t(w_t) \rangle \\ = & \sum_{t=1}^{T} \langle w_t - \tilde{w}_{t+1}, g_t - \tilde{m}_t \rangle + \langle w_t - \tilde{w}_{t+1}, \tilde{m}_t \rangle + \langle \tilde{w}_{t+1} - w^*, \tilde{g}_t \rangle + \langle \tilde{w}_{t+1} - w^*, g_t - \tilde{g}_t \rangle .\end{aligned} \tag{15}$$

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

$$\langle \hat{w} - u, \theta \rangle \leq B_\psi(u, v) - B_\psi(u, \hat{w}) - B_\psi(\hat{w}, v) \quad \text{for any } u \in \Theta . \tag{16}$$

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

$$\tilde{w}_{t+1} = \arg\min_{w \in \Theta} \eta_t \langle w, \tilde{g}_t \rangle + B_{\psi_t}(w, \tilde{w}_t) , \tag{17}$$

By using (16) for (17) with $\hat{w} = \tilde{w}_{t+1}$ (the output of the minimization problem), $u = w^*$ and $v = \tilde{w}_t$, we have

$$\langle \tilde{w}_{t+1} - w^*, \tilde{g}_t \rangle \leq \frac{1}{\eta_t} \left[B_{\psi_t}(w^*, \tilde{w}_t) - B_{\psi_t}(w^*, \tilde{w}_{t+1}) - B_{\psi_t}(\tilde{w}_{t+1}, \tilde{w}_t) \right] . \tag{18}$$

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

$$w_{t+1} = \arg\min_{w \in \Theta} \eta_{t+1} \langle w, \tilde{m}_{t+1} \rangle + B_{\psi_t}(w, \tilde{w}_{t+1}). \tag{19}$$

and, by using (16) for (19) (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

$$\langle w_t - \tilde{w}_{t+1}, \tilde{m}_t \rangle \leq \frac{1}{\eta_t} \left[B_{\psi_{t-1}}(\tilde{w}_{t+1}, \tilde{w}_t) - B_{\psi_{t-1}}(\tilde{w}_{t+1}, w_t) - B_{\psi_{t-1}}(w_t, \tilde{w}_t) \right] , \tag{20}$$

By (15), (18), and (20), we obtain

$$\mathcal{R}_T \overset{(15)}{\leq} \sum_{t=1}^{T} \langle w_t - \tilde{w}_{t+1}, g_t - \tilde{m}_t\rangle + \langle w_t - \tilde{w}_{t+1}, \tilde{m}_t\rangle + \langle \tilde{w}_{t+1} - w^*, \tilde{g}_t\rangle + \langle \tilde{w}_{t+1} - w^*, g_t - \tilde{g}_t\rangle$$

$$\overset{(18),(20)}{\leq} \sum_{t=1}^{T} \|w_t - \tilde{w}_{t+1}\|_{\psi_{t-1}} \|g_t - \tilde{m}_t\|_{\psi^*_{t-1}} + \|\tilde{w}_{t+1} - w^*\|_{\psi_{t-1}} \|g_t - \tilde{g}_t\|_{\psi^*_{t-1}}$$

$$+ \frac{1}{\eta_t}\big[B_{\psi_{t-1}}(\tilde{w}_{t+1}, \tilde{w}_t) - B_{\psi_{t-1}}(\tilde{w}_{t+1}, w_t) - B_{\psi_{t-1}}(w_t, \tilde{w}_t) + B_{\psi_t}(w^*, \tilde{w}_t) - B_{\psi_t}(w^*, \tilde{w}_{t+1}) - B_{\psi_t}(\tilde{w}_{t+1}, \tilde{w}_t)\big], \tag{21}$$

which is further bounded by

$$\mathcal{R}_T \leq \sum_{t=1}^{T} \Big\{ \frac{1}{2\eta_t}\|w_t - \tilde{w}_{t+1}\|^2_{\psi_{t-1}} + \frac{\eta_t}{2}\|g_t - m_t\|^2_{\psi^*_{t-1}} + \|\tilde{w}_{t+1} - w^*\|_{\psi_{t-1}} \|g_t - \tilde{g}_t\|_{\psi^*_{t-1}}$$
$$+ \frac{1}{\eta_t}\Big(\underbrace{B_{\psi_{t-1}}(\tilde{w}_{t+1}, \tilde{w}_t) - B_{\psi_t}(\tilde{w}_{t+1}, \tilde{w}_t)}_{A_1} - \frac{1}{2}\|\tilde{w}_{t+1} - w_t\|^2_{\psi_{t-1}} + \underbrace{B_{\psi_t}(w^*, \tilde{w}_t) - B_{\psi_t}(w^*, \tilde{w}_{t+1})}_{A_2} \Big) \Big\} , \tag{22}$$

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

To proceed, notice that

$$A_1 = B_{\psi_{t-1}}(\tilde{w}_{t+1}, \tilde{w}_t) - B_{\psi_t}(\tilde{w}_{t+1}, \tilde{w}_t) = \langle \tilde{w}_{t+1} - \tilde{w}_t, \mathrm{diag}(\hat{v}^{1/2}_{t-1} - \hat{v}^{1/2}_t)(\tilde{w}_{t+1} - \tilde{w}_t)\rangle \leq 0 , \tag{23}$$

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

$$\begin{aligned} A_2 = B_{\psi_t}(w^*, \tilde{w}_t) - B_{\psi_t}(w^*, \tilde{w}_{t+1}) &= \langle w^* - \tilde{w}_{t+1}, \mathrm{diag}(\hat{v}^{1/2}_{t+1} - \hat{v}^{1/2}_t)(w^* - \tilde{w}_{t+1})\rangle \\ &\leq (\max_i (w^*[i] - \tilde{w}_{t+1}[i])^2) \cdot (\sum_{i=1}^{d} \hat{v}^{1/2}_{t+1}[i] - \hat{v}^{1/2}_t[i]) \end{aligned} \tag{24}$$

Therefore, by (22),(24),(23), we have

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

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.

$\square$

B Proofs of Auxiliary Lemmas

B.1 Proof of Lemma 1

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

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

Proof Assume assumption H4 we have:

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

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

$$\|\theta_{t+1}\| = \|\beta_1\theta_t + (1-\beta_1)\, g_{t+1}\| \leq \beta_1 \|\theta_t\| + (1-\beta_1)\, \|g_{t+1}\| \leq \mathsf{M} \tag{25}$$

Using the same induction reasoning we prove that

$$\|\hat{v}_{t+1}\| = \left\|\beta_2\hat{v}_t + (1-\beta_2)\, g_{t+1}^2\right\| \leq \beta_2 \|\hat{v}_t\| + (1-\beta_1)\, \left\|g_{t+1}^2\right\| \leq \mathsf{M}^2 \tag{26}$$

□

B.2 Proof of Lemma 2

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

$$\overline{w}_{t+1} - \overline{w}_t \leq \frac{\beta_1}{1-\beta_1}\tilde{\theta}_{t-1}\left[\eta_{t-1}\hat{v}_{t-1}^{-1/2} - \eta_t\hat{v}_t^{-1/2}\right] - \eta_t\hat{v}_t^{-1/2}\tilde{g}_t\ , \tag{27}$$

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

Proof By definition (7) and using the Algorithm updates, we have:

$$\begin{aligned}
\overline{w}_{t+1} - \overline{w}_t &= \frac{1}{1-\beta_1}(w_{t+1} - \tilde{w}_t) - \frac{\beta_1}{1-\beta_1}(w_t - \tilde{w}_{t-1}) \\
&= -\frac{1}{1-\beta_1}\eta_t\hat{v}_t^{-1/2}(\theta_t + h_{t+1}) + \frac{\beta_1}{1-\beta_1}\eta_{t-1}\hat{v}_{t-1}^{-1/2}(\theta_{t-1} + h_t) \\
&= -\frac{1}{1-\beta_1}\eta_t\hat{v}_t^{-1/2}(\theta_t + \beta_1\theta_{t-1}) - \frac{1}{1-\beta_1}\eta_t\hat{v}_t^{-1/2}(1-\beta_1)m_{t+1} \\
&+ \frac{\beta_1}{1-\beta_1}\eta_{t-1}\hat{v}_{t-1}^{-1/2}(\theta_{t-1} + \beta_1\theta_{t-2}) + \frac{\beta_1}{1-\beta_1}\eta_{t-1}\hat{v}_{t-1}^{-1/2}(1-\beta_1)m_t
\end{aligned} \tag{28}$$

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

$$\overline{w}_{t+1} - \overline{w}_t \leq \frac{\beta_1}{1-\beta_1}\tilde{\theta}_{t-1}\left[\eta_{t-1}\hat{v}_{t-1}^{-1/2} - \eta_t\hat{v}_t^{-1/2}\right] - \eta_t\hat{v}_t^{-1/2}\tilde{g}_t \tag{29}$$

□

B.3 Proof of Lemma 3

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

$$\sum_{t=1}^{T_{\mathsf{max}}} \eta_t^2 \mathbb{E}\left[\left\|\hat{v}_t^{-1/2}\theta_t\right\|_2^2\right] \leq \frac{\eta^2 d T_{\mathsf{max}}(1-\beta_1)}{(1-\beta_2)(1-\gamma)} \tag{30}$$

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

$$\begin{aligned}
\eta_t^2 \mathbb{E}\left[\left\|\hat{v}_t^{-1/2}\theta_t\right\|_2^2\right] &= \eta_t^2 \mathbb{E}\left[\sum_{p=1}^d \frac{\theta_{t,p}^2}{\hat{v}_{t,p}}\right] \\
&\leq \eta_t^2 \mathbb{E}\left[\sum_{i=1}^d \frac{\theta_{t,p}^2}{v_{t,p}}\right] \\
&\leq \eta_t^2 \mathbb{E}\left[\sum_{i=1}^d \frac{(\sum_{r=1}^t (1-\beta_1)\beta_1^{t-r} g_{r,p})^2}{\sum_{r=1}^t (1-\beta_2)\beta_2^{t-r} g_{r,p}^2}\right]
\end{aligned} \tag{31}$$

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

$$\begin{aligned}
\eta_t^2 \mathbb{E}\left[\left\|\hat{v}_t^{-1/2}\theta_t\right\|_2^2\right] &\leq \frac{\eta_t^2(1-\beta_1)^2}{1-\beta_2} \mathbb{E}\left[\sum_{i=1}^d \frac{(\sum_{r=1}^t \beta_1^{t-r} g_{r,p})^2}{\sum_{r=1}^t \beta_2^{t-r} g_{r,p}^2}\right] \\
&\overset{(a)}{\leq} \frac{\eta_t^2(1-\beta_1)}{1-\beta_2} \mathbb{E}\left[\sum_{i=1}^d \frac{\sum_{r=1}^t \beta_1^{t-r} g_{r,p}^2}{\sum_{r=1}^t \beta_2^{t-r} g_{r,p}^2}\right] \\
&\leq \frac{\eta_t^2(1-\beta_1)}{1-\beta_2} \mathbb{E}\left[\sum_{i=1}^d \sum_{r=1}^t \gamma^{t-r}\right] = \frac{\eta_t^2 d(1-\beta_1)}{1-\beta_2} \mathbb{E}\left[\sum_{r=1}^t \gamma^{t-r}\right]
\end{aligned} \tag{32}$$

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

$$\begin{aligned}
\sum_{t=1}^{T_{\mathsf{max}}} \eta_t^2 \mathbb{E}\left[\left\|\hat{v}_t^{-1/2}\theta_t\right\|_2^2\right] &\leq \frac{\eta_t^2 d(1-\beta_1)}{1-\beta_2} \mathbb{E}\left[\sum_{t=1}^{T_{\mathsf{max}}} \sum_{r=1}^t \gamma^{t-r}\right] \\
&\leq \frac{\eta^2 dT(1-\beta_1)}{1-\beta_2} \mathbb{E}\left[\sum_{t=t}^t \gamma^{t-r}\right] \\
&\leq \frac{\eta^2 dT(1-\beta_1)}{(1-\beta_2)(1-\gamma)}
\end{aligned} \tag{33}$$

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

C Proof of Theorem 2

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

$$\mathbb{E}\left[\|\nabla f(w_T)\|^2\right] \leq \tilde{C}_1 \sqrt{\frac{d}{T_{\mathsf{max}}}} + \tilde{C}_2 \frac{1}{T_{\mathsf{max}}} \tag{34}$$

where T is a random termination number distributed according (6) and the constants are defined as follows:

$$\begin{aligned}
\tilde{C}_1 &= C_1 + \frac{\mathsf{M}}{(1-a\beta_1)+(\beta_1+a)}\left[\frac{a(1-\beta_1)^2}{1-\beta_2} + 2L\frac{1}{1-\beta_2}\right] \\
C_1 &= \frac{\mathsf{M}}{(1-a\beta_1)+(\beta_1+a)}\Delta f + \frac{4L\left(\frac{\beta_1}{1-\beta_1}\right)^2 \mathsf{M}}{(1-a\beta_1)+(\beta_1+a)} \frac{(1+\beta_1^2)(1-\beta_1)}{(1-\beta_2)(1-\gamma)} \\
\tilde{C}_2 &= \frac{\mathsf{M}}{(1-\beta_1)\left((1-a\beta_1)+(\beta_1+a)\right)} \tilde{\mathsf{M}}^2 \mathbb{E}\left[\left\|\hat{v}_0^{-1/2}\right\|\right]
\end{aligned} \tag{35}$$

Proof Using H2 and the iterate $\overline{w}_t$ we have:

$$\begin{aligned} f(\overline{w}_{t+1}) &\leq f(\overline{w}_t) + \nabla f(\overline{w}_t)^\top (\overline{w}_{t+1} - \overline{w}_t) + \frac{L}{2} \|\overline{w}_{t+1} - \overline{w}_t\|^2 \\ &\leq f(\overline{w}_t) + \underbrace{\nabla f(w_t)^\top (\overline{w}_{t+1} - \overline{w}_t)}_{A} + \underbrace{\left(\nabla f(\overline{w}_t) - \nabla f(w_t)\right)^\top (\overline{w}_{t+1} - \overline{w}_t)}_{B} + \frac{L}{2} \|\overline{w}_{t+1} - \overline{w}_t\| \end{aligned} \tag{36}$$

Term A. Using Lemma 2, we have that:

$$\begin{aligned} \nabla f(w_t)^\top (\overline{w}_{t+1} - \overline{w}_t) &\leq \nabla f(w_t)^\top \left[\frac{\beta_1}{1-\beta_1} \tilde{\theta}_{t-1} \left[\eta_{t-1} \hat{v}_{t-1}^{-1/2} - \eta_t \hat{v}_t^{-1/2} \right] - \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \right] \\ &\leq \frac{\beta_1}{1-\beta_1} \|\nabla f(w_t)\| \left\| \eta_{t-1} \hat{v}_{t-1}^{-1/2} - \eta_t \hat{v}_t^{-1/2} \right\| \left\| \tilde{\theta}_{t-1} \right\| - \nabla f(w_t)^\top \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \end{aligned} \tag{37}$$

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

$$\nabla f(w_t)^\top (\overline{w}_{t+1} - \overline{w}_t) \leq \frac{\beta_1(1+\beta_1)}{1-\beta_1} \mathsf{M}^2 \left[\left\| \eta_{t-1} \hat{v}_{t-1}^{-1/2} \right\| - \left\| \eta_t \hat{v}_t^{-1/2} \right\| \right] - \nabla f(w_t)^\top \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \tag{38}$$

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

$$\begin{aligned} -\nabla f(w_t)^\top \eta_t \hat{v}_t^{-1/2} \tilde{g}_t &= -\nabla f(w_t)^\top \eta_{t-1} \hat{v}_{t-1}^{-1/2} \bar{g}_t - \nabla f(w_t)^\top \left[\eta_t \hat{v}_t^{-1/2} - \eta_t \hat{v}_t^{-1/2} \right] \bar{g}_t \\ &\quad - \nabla f(w_t)^\top \eta_{t-1} \hat{v}_{t-1}^{-1/2} (\beta_1 g_{t-1} + m_{t+1}) \\ &\leq -\nabla f(w_t)^\top \eta_{t-1} \hat{v}_{t-1}^{-1/2} \bar{g}_t + (1 - a\beta_1) \mathsf{M}^2 \left[\left\| \eta_{t-1} \hat{v}_{t-1}^{-1/2} \right\| - \left\| \eta_t \hat{v}_t^{-1/2} \right\| \right] \\ &\quad - \nabla f(w_t)^\top \eta_t \hat{v}_t^{-1/2} (\beta_1 g_{t-1} + m_{t+1}) \end{aligned} \tag{39}$$

using Lemma 1 on $\|g_t\|$ and where that $\tilde{g}_t = \bar{g}_t + \beta_1 g_{t-1} + m_{t+1} = g_t - \beta_1 m_t + \beta_1 g_{t-1} + m_{t+1}$. Plugging (39) into (38) yields:

$$\begin{aligned} &\nabla f(w_t)^\top (\overline{w}_{t+1} - \overline{w}_t) \\ &\leq -\nabla f(w_t)^\top \eta_{t-1} \hat{v}_{t-1}^{-1/2} \bar{g}_t + \frac{1}{1-\beta_1} (a\beta_1^2 - 2a\beta_1 + \beta 1) \mathsf{M}^2 \left[\left\| \eta_{t-1} \hat{v}_{t-1}^{-1/2} \right\| - \left\| \eta_t \hat{v}_t^{-1/2} \right\| \right] \\ &- \nabla f(w_t)^\top \eta_t \hat{v}_t^{-1/2} (\beta_1 g_{t-1} + m_{t+1}) \end{aligned} \tag{40}$$

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

$$\left(\nabla f(\overline{w}_t) - \nabla f(w_t)\right)^\top (\overline{w}_{t+1} - \overline{w}_t) \leq \|\nabla f(\overline{w}_t) - \nabla f(w_t)\| \, \|\overline{w}_{t+1} - \overline{w}_t\| \tag{41}$$

Using smoothness assumption H2:

$$\begin{aligned} \|\nabla f(\overline{w}_t) - \nabla f(w_t)\| &\leq L \|\overline{w}_t - w_t\| \\ &\leq L \frac{\beta_1}{1-\beta_1} \|w_t - \tilde{w}_{t-1}\| \end{aligned} \tag{42}$$

By Lemma 2 we also have:

$$\begin{aligned} \overline{w}_{t+1} - \overline{w}_t &= \frac{\beta_1}{1-\beta_1} \tilde{\theta}_{t-1} \left[\eta_{t-1} \hat{v}_{t-1}^{-1/2} - \eta_t \hat{v}_t^{-1/2} \right] - \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \\ &= \frac{\beta_1}{1-\beta_1} \tilde{\theta}_{t-1} \eta_{t-1} \hat{v}_{t-1}^{-1/2} \left[I - (\eta_t \hat{v}_t^{-1/2})(\eta_{t-1} \hat{v}_{t-1}^{-1/2})^{-1} \right] - \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \\ &= \frac{\beta_1}{1-\beta_1} \left[I - (\eta_t \hat{v}_t^{-1/2})(\eta_{t-1} \hat{v}_{t-1}^{-1/2})^{-1} \right] (\tilde{w}_{t-1} - w_t) - \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \end{aligned} \tag{43}$$

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:

$$\|\overline{w}_{t+1} - \overline{w}_t\| \leq \frac{\beta_1}{1-\beta_1} \|\tilde{w}_{t-1} - w_t\| + \left\| \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \right\| \tag{44}$$

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

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

Plugging (42) and (44) into (41) returns:

$$\begin{aligned}(\nabla f(\overline{w}_t) - \nabla f(w_t))^\top (\overline{w}_{t+1} - \overline{w}_t) \leq & L \frac{\beta_1}{1-\beta_1} \left\| \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \right\| \|w_t - \tilde{w}_{t-1}\| \\ & + L \left(\frac{\beta_1}{1-\beta_1} \right)^2 \|\tilde{w}_{t-1} - w_t\|^2 \end{aligned} \tag{45}$$

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

$$(\nabla f(\overline{w}_t) - \nabla f(w_t))^\top (\overline{w}_{t+1} - \overline{w}_t) \leq L \left\| \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \right\|^2 + 2L \left(\frac{\beta_1}{1-\beta_1} \right)^2 \|\tilde{w}_{t-1} - w_t\|^2 \tag{46}$$

The last term $\frac{L}{2} \|\overline{w}_{t+1} - \overline{w}_t\|$ can be upper bounded using (44):

$$\begin{aligned}\frac{L}{2} \|\overline{w}_{t+1} - \overline{w}_t\|^2 & \leq \frac{L}{2} \left[\frac{\beta_1}{1-\beta_1} \|\tilde{w}_{t-1} - w_t\| + \left\| \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \right\| \right] \\ & \leq L \left\| \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \right\|^2 + 2L \left(\frac{\beta_1}{1-\beta_1} \right)^2 \|\tilde{w}_{t-1} - w_t\|^2 \end{aligned} \tag{47}$$

Plugging (40), (46) and (47) into (36) and taking the expectations on both sides give:

$$\begin{aligned}&\mathbb{E}\left[f(\overline{w}_{t+1}) + \frac{1}{1-\beta_1} \tilde{\mathsf{M}}^2 \left\| \eta_t \hat{v}_t^{-1/2} \right\| - \left(f(\overline{w}_t) + \frac{1}{1-\beta_1} \tilde{\mathsf{M}}^2 \left\| \eta_{t-1} \hat{v}_{t-1}^{-1/2} \right\| \right) \right] \\ &\leq \mathbb{E}\left[-\nabla f(w_t)^\top \eta_{t-1} \hat{v}_{t-1}^{-1/2} \overline{g}_t - \nabla f(w_t)^\top \eta_t \hat{v}_t^{-1/2} (\beta_1 g_{t-1} + m_{t+1}) \right] \\ &+ \mathbb{E}\left[2L \left\| \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \right\|^2 + 4L \left(\frac{\beta_1}{1-\beta_1} \right)^2 \|\tilde{w}_{t-1} - w_t\|^2 \right] \end{aligned} \tag{48}$$

where $\tilde{\mathsf{M}}^2 = (a\beta_1^2 - 2a\beta_1 + \beta 1)\mathsf{M}^2$. Note that the expectation of $\tilde{g}_t$ conditioned on the filtration $\mathcal{F}_t$ reads as follows

$$\begin{aligned}\mathbb{E}\left[\nabla f(w_t)^\top \overline{g}_t \right] &= \mathbb{E}\left[\nabla f(w_t)^\top (g_t - \beta_1 m_t) \right] \\ &= (1 - a\beta_1) \|\nabla f(w_t)\|^2 \end{aligned} \tag{49}$$

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

$$\begin{aligned}&\frac{1}{\mathsf{M}} \sum_{t=1}^{T_{\max}} ((1 - a\beta_1)\eta_{t-1} + (\beta_1 + a)\eta_t) \|\nabla f(w_t)\|^2 \leq \\ &\mathbb{E}\left[f(\overline{w}_1) + \frac{1}{1-\beta_1} \tilde{\mathsf{M}}^2 \left\| \eta_0 \hat{v}_0^{-1/2} \right\| - \left(f(\overline{w}_{T_{\max}+1}) + \frac{1}{1-\beta_1} \tilde{\mathsf{M}}^2 \left\| \eta_{T_{\max}} \hat{v}_{T_{\max}}^{-1/2} \right\| \right) \right] \\ &+ 2L \sum_{t=1}^{T_{\max}} \mathbb{E}\left[\left\| \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \right\|^2 \right] + 4L \left(\frac{\beta_1}{1-\beta_1} \right)^2 \sum_{t=1}^{T_{\max}} \mathbb{E}\left[\|\tilde{w}_{t-1} - w_t\|^2 \right] \\ &\leq \mathbb{E}\left[\Delta f + \frac{1}{1-\beta_1} \tilde{\mathsf{M}}^2 \left\| \eta_0 \hat{v}_0^{-1/2} \right\| \right] + 2L \sum_{t=1}^{T_{\max}} \mathbb{E}\left[\left\| \eta_t \hat{v}_t^{-1/2} \tilde{g}_t \right\|^2 \right] + 4L \left(\frac{\beta_1}{1-\beta_1} \right)^2 \sum_{t=1}^{T_{\max}} \mathbb{E}\left[\|\tilde{w}_{t-1} - w_t\|^2 \right] \end{aligned} \tag{50}$$

where $\Delta f = f(\overline{w}_1) - f(\overline{w}_{T_{\max}+1})$. We note that by definition of $\hat{v}_t$, and a constant learning rate η_t, we have

$$
\begin{aligned}
\|\tilde{w}_{t-1} - w_t\|^2 &= \left\|\eta_{t-1}\hat{v}_{t-1}^{-1/2}(\theta_{t-1} + h_t)\right\|^2 \\
&= \left\|\eta_{t-1}\hat{v}_{t-1}^{-1/2}(\theta_{t-1} + \beta_1\theta_{t-2} + (1-\beta_1)m_t)\right\|^2 \\
&\leq \left\|\eta_{t-1}\hat{v}_{t-1}^{-1/2}\theta_{t-1}\right\|^2 + \left\|\eta_{t-2}\hat{v}_{t-2}^{-1/2}\beta_1\theta_{t-2}\right\|^2 + (1-\beta_1)^2\left\|\eta_{t-1}\hat{v}_{t-1}^{-1/2}m_t\right\|^2
\end{aligned}
\tag{51}
$$

Using Lemma 3 we have

$$
\begin{aligned}
&\sum_{t=1}^{T_{\max}} \mathbb{E}\left[\|\tilde{w}_{t-1} - w_t\|^2\right] \\
&\leq (1+\beta_1^2)\frac{\eta^2 d T_{\max}(1-\beta_1)}{(1-\beta_2)(1-\gamma)} + (1-\beta_1)^2 \sum_{t=1}^{T_{\max}} \mathbb{E}\left[\left\|\eta_{t-1}\hat{v}_{t-1}^{-1/2}m_t\right\|^2\right]
\end{aligned}
\tag{52}
$$

And thus, setting the learning rate to a constant value η and injecting in (50) yields:

$$
\begin{aligned}
\mathbb{E}\left[\|\nabla f(w_T)\|^2\right] &= \frac{1}{\sum_{j=1}^{T_{\max}}\eta_j}\sum_{t=1}^{T_{\max}}\eta_t\|\nabla f(w_t)\|^2 \\
&\leq \frac{\mathsf{M}}{(1-a\beta_1)+(\beta_1+a)}\frac{1}{\sum_{j=1}^{T_{\max}}\eta_j}\mathbb{E}\left[\Delta f + \frac{1}{1-\beta_1}\tilde{\mathsf{M}}^2\left\|\eta_0\hat{v}_0^{-1/2}\right\|\right] \\
&+ \frac{4L\left(\frac{\beta_1}{1-\beta_1}\right)^2\mathsf{M}}{(1-a\beta_1)+(\beta_1+a)}\frac{1}{\sum_{j=1}^{T_{\max}}\eta_j}(1+\beta_1^2)\frac{\eta^2 d T_{\max}(1-\beta_1)}{(1-\beta_2)(1-\gamma)} \\
&+ \frac{\mathsf{M}}{(1-a\beta_1)+(\beta_1+a)}\frac{1}{\sum_{j=1}^{T_{\max}}\eta_j}(1-\beta_1)^2\sum_{t=1}^{T_{\max}}\mathbb{E}\left[\left\|\eta_{t-1}\hat{v}_{t-1}^{-1/2}m_t\right\|^2\right] \\
&+ \frac{2L\mathsf{M}}{(1-a\beta_1)+(\beta_1+a)}\frac{1}{\sum_{j=1}^{T_{\max}}\eta_j}\sum_{t=1}^{T_{\max}}\mathbb{E}\left[\left\|\eta_t\hat{v}_t^{-1/2}\tilde{g}_t\right\|^2\right]
\end{aligned}
\tag{53}
$$

where T is a random termination number distributed according (6). Setting the stepsize to $\eta = \frac{1}{\sqrt{dT_{\max}}}$ yields :

$$
\begin{aligned}
&\mathbb{E}\left[\|\nabla f(w_T)\|^2\right] \\
&\leq C_1\sqrt{\frac{d}{T_{\max}}} + C_2\frac{1}{T_{\max}} \\
&+ D_1\frac{\eta}{T_{\max}}\sum_{t=1}^{T_{\max}}\mathbb{E}\left[\left\|\hat{v}_{t-1}^{-1/2}m_t\right\|^2\right] + D_2\frac{\eta}{T_{\max}}\sum_{t=1}^{T_{\max}}\mathbb{E}\left[\left\|\hat{v}_{t-1}^{-1/2}\tilde{g}_t\right\|^2\right]
\end{aligned}
\tag{54}
$$

where

$$
\begin{aligned}
C_1 &= \frac{\mathsf{M}}{(1-a\beta_1)+(\beta_1+a)}\Delta f + \frac{4L\left(\frac{\beta_1}{1-\beta_1}\right)^2\mathsf{M}}{(1-a\beta_1)+(\beta_1+a)}\frac{(1+\beta_1^2)(1-\beta_1)}{(1-\beta_2)(1-\gamma)} \\
C_2 &= \frac{\mathsf{M}}{(1-\beta_1)\left((1-a\beta_1)+(\beta_1+a)\right)}\tilde{\mathsf{M}}^2\mathbb{E}\left[\left\|\hat{v}_0^{-1/2}\right\|\right]
\end{aligned}
\tag{55}
$$

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 that:

$$
\sum_{t=1}^{T_{\max}}\eta_t^2\mathbb{E}\left[\left\|\hat{v}_t^{-1/2}g_t\right\|_2^2\right] \leq \frac{\eta^2 d T_{\max}}{(1-\beta_2)}
\tag{56}
$$

which leads to the final bound:

$$\begin{aligned}&\mathbb{E}\left[\|\nabla f(w_T)\|^2\right]\\&\leq \tilde{C}_1\sqrt{\frac{d}{T_{\mathsf{max}}}}+\tilde{C}_2\frac{1}{T_{\mathsf{max}}}\end{aligned}\tag{57}$$

where

$$\begin{aligned}\tilde{C}_1 &= C_1+\frac{\mathsf{M}}{(1-a\beta_1)+(\beta_1+a)}\left[\frac{a(1-\beta_1)^2}{1-\beta_2}+2L\frac{1}{1-\beta_2}\right]\\ \tilde{C}_2 &= C_2=\frac{\mathsf{M}}{(1-\beta_1)\left((1-a\beta_1)+(\beta_1+a)\right)}\tilde{\mathsf{M}}^2\mathbb{E}\left[\left\|\hat{v}_0^{-1/2}\right\|\right]\end{aligned}\tag{58}$$

□

D Proof of Lemma 4 (Boundedness of the iterates)

Lemma. *Given the multilayer model (9), assume the boundedness of the input data and of the loss function,* i.e., *for any $\xi \in \mathbb{R}^l$ and $y \in \mathbb{R}$ there is a constant $T > 0$ such that:*

$$\|\xi\| \leq 1 \quad a.s. \quad and |\mathcal{L}'(\cdot, y)| \leq T \tag{59}$$

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:*

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

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

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

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

$$\begin{aligned}\|\nabla_{w^{(L)}}\mathcal{L}(\mathsf{MLN}(w,\xi),y)\| &= \|\mathcal{L}'(\mathsf{MLN}(w,\xi),y)\nabla_{w^{(L)}}\mathsf{MLN}(w,\xi)\|\\ &= \left\|\mathcal{L}'(\mathsf{MLN}(w,\xi),y)\sigma'(w^{(L)}h^{(L-1)}(w,\xi))h^{(L-1)}(w,\xi)\right\|\\ &\leq \frac{T}{4}\end{aligned}\tag{60}$$

where the last equality is due to mild assumptions (59) and to the fact that the norm of the derivative of the sigmoid function is upperbounded by $1/4$.

From Algorithm 2, with $\beta_1 = 0$ we have for iteration index $t > 0$:

$$\begin{aligned}\|w_t-\tilde{w}_{t-1}\| &= \left\|-\eta_t\hat{v}_t^{-1/2}(\theta_t+h_{t+1})\right\|\\ &= \left\|\eta_t\hat{v}_t^{-1/2}(g_t+m_{t+1})\right\|\\ &\leq \hat{\eta}\left\|\hat{v}_t^{-1/2}g_t\right\|+\hat{\eta}a\left\|\hat{v}_t^{-1/2}g_{t+1}\right\|\end{aligned}\tag{61}$$

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

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

. Thus:

$$\begin{aligned}\|w_t-\tilde{w}_{t-1}\| &\leq \hat{\eta}\left(\left\|\hat{v}_t^{-1/2}g_t\right\|+a\left\|\hat{v}_t^{-1/2}g_{t+1}\right\|\right)\\ &\leq \hat{\eta}\frac{a+1}{\sqrt{1-\beta_2}}\end{aligned}\tag{62}$$

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 (60), we have

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

where $f(\cdot)$ is defined by (10) and is the loss of our MLN. This last equation yields $\theta_{T,i}^{(L)} \geq 0$ (given the algorithm and $\beta_1 = 0$) and using the fact that $\|w_t - \tilde{w}_{t-1}\| \leq B$ we have

$$0 \leq w_{T-1,i}^{(L)} - B \leq w_{T,i}^{(L)} \leq w_{T-1,i}^{(L)} \tag{63}$$

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 (63) 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

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

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

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

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

$$\nabla_{w^{(\ell)}} \mathcal{L}(\mathsf{MLN}(w,\xi), y) = \mathcal{L}'(\mathsf{MLN}(w,\xi), y) \left(\prod_{j=1}^{\ell+1} \sigma' \left(w^{(j)} h^{(j-1)}(w,\xi) \right) \right) h^{(\ell-1)}(w,\xi) \tag{64}$$

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 $\left\|w^{(\ell)}\right\|_F \leq F_\ell$ with the Frobenius norm. Suppose we have shown $\left\|w^{(r)}\right\|_F \leq F_r$ for any layer $r > \ell$. Then having this gradient (64) 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

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

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

E Comparison to some related methods

Comparison to nonconvex optimization works. Recently, [40, 5, 38, 42, 44, 21] provide some theoretical analysis of ADAM-type algorithms when applying them to smooth nonconvex optimization problems. For example, [5] provides a bound, which is $\min_{t\in[T]} \mathbb{E}[\|\nabla f(w_t)\|^2] = O(\log T/\sqrt{T})$. 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 OPTIMISTIC-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 non-convex loss functions. Their approaches are modular so that simply using OPTIMISTIC-AMSGRAD as the base algorithm in their methods will immediately lead to a variant of OPTIMISTIC-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 $r_{0:t}(w) := \langle w, \mathrm{diag}\{\hat{v}_t\}^{1/2} w\rangle$ and $\|\cdot\|_{(t)} := \sqrt{\langle \cdot, \mathrm{diag}\{\hat{v}_t\}^{1/2} \cdot\rangle}$, 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 β_1, β_2, and η_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 [14]), while the proposed algorithm in this paper is designed to accelerate optimization (e.g. solving empirical risk minimization quickly). [8] focuses on training GANs [14]. 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.

F Additional Remarks and Runs on the Gradient Prediction Process

Two illustrative examples. We provide two toy examples to demonstrate how OPTIMISTIC-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 $g_t := \nabla H(w_t)$ 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, the update can be written in the form of (12) with $A = (1 - b\eta)$ and $u_{t-1} = 0$ by setting $\eta_t = \eta$ (constant step size). Therefore, the extrapolation method should predict well.

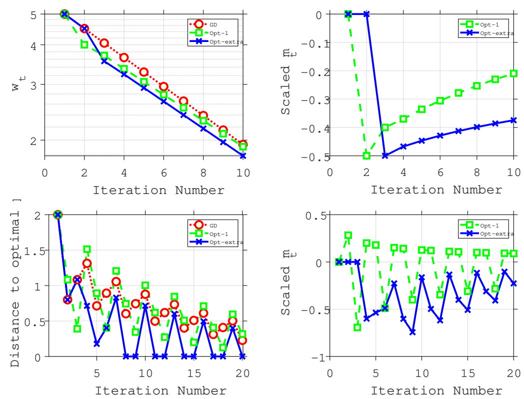

Figure 5: (a): The iterate w_t; the closer to the optimal point 0 the better. (b): A scaled and clipped version of m_t: $w_t - w_{t-1/2}$, which measures how the prediction of m_t drives the update towards the optimal point. In this scenario, the more negative the better. (c): Distance to the optimal 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.

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

Now let us consider another problem: an online learning problem proposed in [31] [3]. Assume the learner's decision space is $\Theta = [-1, 1]$, and the loss function is $\ell_t(w) = 3w$ if $t \bmod 3 = 1$, and $\ell_t(w) = -w$ otherwise. The optimal point to minimize the cumulative loss is $w^* = -1$. We let $\eta_t = 0.1/\sqrt{t}$ and the initial point $w_0 = 1$ for all the three methods. The parameter λ of the extrapolation method is set to $\lambda = 10^{-3} > 0$. The results are on Figure 5 (c) and (d). Sub-figure (c) shows that Opt-extra converges faster than the other methods while Opt-1 is not better than GD. The reason is that the gradient changes from -1 to 3 at $t \bmod 3 = 1$ and it changes from 3 to -1 at $t \bmod 3 = 2$. Consequently, using the current gradient as the guess for the next clearly is not a good choice, since the next gradient is in the opposite direction of the current one. Sub-figure (d) shows that $-m_t$ by the extrapolation method always points to $w^* = -1$, while the one by using the previous negative direction points to the opposite direction in two thirds of rounds. It shows that the extrapolation method is much less affected by the gradient oscillation and always makes the prediction in the right direction, which suggests that the method can capture the aggregate effect.

[2] The extrapolation method needs at least two gradients for prediction. This is why in the first two iterations, m_t is 0.

[3] [31] uses this example to show that ADAM [18] fails to converge.