

---

# Online Adaptive Methods, Universality and Acceleration

---

**Kfir Y. Levy**  
ETH Zurich

yehuda.levy@inf.ethz.ch

**Alp Yurtsever**  
EPFL

alp.yurtsever@epfl.ch

**Volkan Cevher**  
EPFL

volkan.cevher@epfl.ch

## Abstract

We present a novel method for convex unconstrained optimization that, *without any modifications*, ensures: **(i)** accelerated convergence rate for smooth objectives, **(ii)** standard convergence rate in the general (non-smooth) setting, and **(iii)** standard convergence rate in the stochastic optimization setting. To the best of our knowledge, this is the first method that *simultaneously* applies to all of the above settings.

At the heart of our method is an adaptive learning rate rule that employs importance weights, in the spirit of adaptive online learning algorithms [13, 21], combined with an update that linearly couples two sequences, in the spirit of [2]. An empirical examination of our method demonstrates its applicability to the above mentioned scenarios and corroborates our theoretical findings.

## 1 Introduction

The accelerated gradient method of Nesterov [25] is one of the cornerstones of modern optimization. Due to its appeal as a computationally efficient and fast method, it has found use in numerous applications including: imaging [9], compressed sensing [15], and deep learning [33], amongst other.

Despite these merits, accelerated methods are less prevalent in Machine Learning due to two major issues: **(i)** acceleration is inappropriate for handling noisy feedback, and **(ii)** acceleration requires the knowledge of the objective's smoothness. While each of these issues was separately resolved in [18, 17, 35], and respectively in [27]; it was unknown whether there exists an accelerated method that addresses both issues. In this work we propose such a method.

Concretely, Nesterov [27] devises a method that obtains an accelerated convergence rate of  $\mathcal{O}(1/T^2)$  for smooth convex objectives, and a standard rate of  $\mathcal{O}(1/\sqrt{T})$  for non-smooth convex objectives, over  $T$  iterations. This is done without any prior knowledge of the smoothness parameter, and is therefore referred to as a *universal*<sup>1</sup> method. Nonetheless, this method uses a line search technique in every round, and is therefore inappropriate for handling noisy feedback. On the other hand, Lan [18], Hu et al. [17], and Xiao [35], devise accelerated methods that are able to handle noisy feedback and obtain a convergence rate of  $\mathcal{O}(1/T^2 + \sigma/\sqrt{T})$ , where  $\sigma$  is the variance of the gradients. However, these methods are not universal since they require the knowledge of both  $\sigma$  and of the smoothness.

Conversely, adaptive first order methods are very popular in Machine Learning, with AdaGrad, [13], being the most prominent method among this class. AdaGrad is an online learning algorithm which adapts its learning rate using the feedback (gradients) received through the optimization process, and is known to successfully handle noisy feedback. This renders AdaGrad as

---

<sup>1</sup>Following Nesterov's paper [27], we say that an algorithm is *universal* if it does not require to know in advance whether the objective is smooth or not. Note that universality does not mean a parameter free algorithm. Specifically, Nesterov's universal methods [27] as well as ours are not parameter free.

the method of choice in various learning applications. Note however, that AdaGrad (probably) can not ensure acceleration. Moreover, it was so far unknown whether AdaGrad is at all able to exploit smoothness in order to converge faster.

In this work we investigate unconstrained convex optimization. We suggest AcceleGrad (Alg. 2), a novel *universal* method which employs an accelerated-gradient-like update rule together with an adaptive learning rate à la AdaGrad. Our contributions,

- We show that AcceleGrad obtains an accelerated rate of  $\mathcal{O}(1/T^2)$  in the smooth case and  $\tilde{\mathcal{O}}(1/\sqrt{T})$  in the general case, without any prior information of the objective’s smoothness.
- We show that *without any modifications*, AcceleGrad ensures a convergence rate of  $\tilde{\mathcal{O}}(1/\sqrt{T})$  in the general stochastic convex case.
- We also present a new result regarding the AdaGrad algorithm. We show that in the case of stochastic optimization with a smooth expected loss, AdaGrad ensures an  $\mathcal{O}(1/T + \sigma/\sqrt{T})$  convergence rate, where  $\sigma$  is the variance of the gradients. AdaGrad does not require a knowledge of the smoothness, hence this result establishes the universality of AdaGrad (though without acceleration).

On the technical side our algorithm employs three simultaneous mechanisms: learning rate adaptation in conjunction with importance weighting, in the spirit of adaptive online learning algorithms [13, 21], combined with an update rule that linearly couples two sequences, in the spirit of [2].

This paper is organized as follows. In Section 2 we present our setup and review relevant background. Our results and analysis for the offline setting are presented in Section 3, and Section 4 presents our results for the stochastic setting. In Section 5 we present our empirical study, and Section 6 concludes.

**Related Work:** In his pioneering work, Nesterov [25], establishes an accelerated rate for smooth convex optimization. This was later generalized in, [26, 6], to allow for general metrics and line search.

In recent years there has been a renewed interest in accelerated methods, with efforts being made to understand acceleration as well as to extend it beyond the standard offline optimization setting.

An extension of acceleration to handle stochastic feedback was developed in, [18, 17, 35, 10]. Acceleration for modern variance reduction optimization methods is explored in, [31, 1], and generic templates to accelerating variance reduction algorithms are developed in, [22, 16]. Scieur et al. [30], derives a scheme that enables hindsight acceleration of non-accelerated methods. In [36], the authors devise a universal accelerated method for primal dual problems. And the connection between acceleration and ODEs is investigated in, [32, 34, 14, 20, 5, 4]. Universal accelerated schemes are explored in [27, 19, 28], yet these works do not apply to the stochastic setting. Alternative accelerated methods and interpretations are explored in, [3, 7, 12].

Curiously, Allen-Zhu and Orecchia [2], interpret acceleration as a linear coupling between gradient descent and mirror descent, our work builds on their ideas. Our method also relies on ideas from [21], where universal (non-accelerated) procedures are derived through a conversion scheme of online learning algorithms.

## 2 Setting and Preliminaries

We discuss the optimization of a convex function  $f : \mathbb{R}^d \mapsto \mathbb{R}$ . Our goal is to (approximately) solve the following unconstrained optimization problem,

$$\min_{x \in \mathbb{R}^d} f(x).$$

We focus on first order methods, i.e., methods that only require gradient information, and consider both smooth and non-smooth objectives. The former is defined below,

**Definition 1** ( $\beta$ -smoothness). *A function  $f : \mathbb{R}^d \mapsto \mathbb{R}$  is  $\beta$ -smooth if,*

$$f(y) \leq f(x) + \nabla f(x) \cdot (y - x) + \frac{\beta}{2} \|x - y\|^2; \quad \forall x, y \in \mathbb{R}^d$$

---

**Algorithm 1** Adaptive Gradient Method (AdaGrad)

---

**Input:** #Iterations  $T$ ,  $x_1 \in \mathcal{K}$ , set  $\mathcal{K}$

**for**  $t = 1 \dots T$  **do**

    Calculate:  $g_t = \nabla f(x_t)$ , and update,  $\eta_t = D \left( 2 \sum_{\tau=1}^t \|g_\tau\|^2 \right)^{-1/2}$

    Update:

$$x_{t+1} = \Pi_{\mathcal{K}}(x_t - \eta_t g_t)$$

**end for**

Output:  $\bar{x}_T = \frac{1}{T} \sum_{t=1}^T x_t$

---

It is well known that with the knowledge of the smoothness parameter,  $\beta$ , one may obtain fast convergence rates by an appropriate adaptation of the update rule. In this work we do not assume any such knowledge; instead we assume to be given a bound on the distance between some initial point,  $x_0$ , and a global minimizer of the objective.

This is formalized as follows: we are given a compact convex set  $\mathcal{K}$  that contains a global minimum of  $f$ , i.e.,  $\exists z \in \mathcal{K}$  such that  $z \in \arg \min_{x \in \mathbb{R}^d} f(x)$ . Thus, for any initial point,  $x_0 \in \mathcal{K}$ , its distance from the global optimum is bounded by the diameter of the set,  $D := \max_{x, y \in \mathcal{K}} \|x - y\|$ . Note that we allow to choose points outside  $\mathcal{K}$ . We also assume that the objective  $f$  is  $G$ -Lipschitz, which translates to a bound of  $G$  on the magnitudes of the (sub)-gradients.

An access to the exact gradients of the objective is not always possible. And in many scenarios we may only access an oracle which provides noisy and unbiased gradient estimates. This *Stochastic Optimization* setting is prevalent in Machine Learning, and we discuss it more formally in Section 4.

**The AdaGrad Algorithm:** The adaptive method presented in this paper is inspired by AdaGrad (Alg. 1), a well known online optimization method which employs an adaptive learning rate. The following theorem states AdaGrad's guarantees<sup>2</sup>, [13],

**Theorem 2.1.** *Let  $\mathcal{K}$  be a convex set with diameter  $D$ . Let  $f$  be a convex function. Then Algorithm 1 guarantees the following error;*

$$f(\bar{x}_T) - \min_{x \in \mathcal{K}} f(x) \leq \sqrt{2D^2 \sum_{t=1}^T \|g_t\|^2 / T}.$$

**Notation:** We denote the Euclidean norm by  $\|\cdot\|$ . Given a compact convex set  $\mathcal{K}$  we denote by  $\Pi_{\mathcal{K}}(\cdot)$  the projection onto the  $\mathcal{K}$ , i.e.  $\forall x \in \mathbb{R}^d$ ,  $\Pi_{\mathcal{K}}(x) = \arg \min_{y \in \mathcal{K}} \|y - x\|^2$ .

### 3 Offline Setting

This section discusses the offline optimization setting where we have an access to the exact gradients of the objective. We present our method in Algorithm 2, and substantiate its universality by providing  $O(1/T^2)$  rate in the smooth case (Thm. 3.1), and a rate of  $O(\sqrt{\log T/T})$  in the general convex case (Thm. 3.2). The analysis for the smooth case appears in Section 3.1 and we defer the proof of the non-smooth case to the Appendix.

AcceleGrad is summarized in Algorithm 2. Inspired by, [2], our method linearly couples between two sequences  $\{z_t\}_t, \{y_t\}_t$  into a sequence  $\{x_{t+1}\}_t$ . Using the gradient,  $g_t = \nabla f(x_{t+1})$ , these sequences are then updated with the same learning rate,  $\eta_t$ , yet with different *reference points* and *gradient magnitudes*. Concretely,  $y_{t+1}$  takes a gradient step starting at  $x_{t+1}$ . Conversely, for  $z_{t+1}$  we scale the gradient by a factor of  $\alpha_t$  and then take a projected gradient step starting at  $z_t$ . Our method finally outputs a weighted average of the  $\{y_{t+1}\}_t$  sequence.

Our algorithm coincides with the method of [2] upon taking  $\eta_t = 1/\beta$  and outputting the last iterate,  $y_T$ , rather than a weighted average; yet this method is not

---

<sup>2</sup>Actually AdaGrad is well known to ensure regret guarantees in the online setting. For concreteness, Thm. 2.1 provides error guarantees in the offline setting.

---

**Algorithm 2** Accelerated Adaptive Gradient Method (AcceleGrad)

---

**Input:** #Iterations  $T$ ,  $x_0 \in \mathcal{K}$ , diameter  $D$ , weights  $\{\alpha_t\}_{t \in [T]}$ , learning rate  $\{\eta_t\}_{t \in [T]}$

Set:  $y_0 = z_0 = x_0$

**for**  $t = 0 \dots T$  **do**

Set  $\tau_t = 1/\alpha_t$

Update:

$$x_{t+1} = \tau_t z_t + (1 - \tau_t) y_t, \quad \text{and define } g_t := \nabla f(x_{t+1})$$

$$z_{t+1} = \Pi_{\mathcal{K}}(z_t - \alpha_t \eta_t g_t)$$

$$y_{t+1} = x_{t+1} - \eta_t g_t$$

**end for**

Output:  $\bar{y}_T \propto \sum_{t=0}^{T-1} \alpha_t y_{t+1}$

---

universal. Below we present our  $\beta$ -independent choice of learning rate and weights,

$$\eta_t = \frac{2D}{\left(G^2 + \sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2\right)^{1/2}} \quad \& \quad \alpha_t = \begin{cases} 1 & 0 \leq t \leq 2 \\ \frac{1}{4}(t+1) & t \geq 3 \end{cases} \quad (1)$$

The learning rate that we suggest adapts similarly to AdaGrad. Differently from AdaGrad we consider the importance weights,  $\alpha_t$ , inside the learning rate rule; an idea that we borrow from [21]. The weights that we employ are increasing with  $t$ , which in turn emphasizes recent queries.

Next we state the guarantees of AcceleGrad for the smooth and non-smooth cases,

**Theorem 3.1.** *Assume that  $f$  is convex and  $\beta$ -smooth. Let  $\mathcal{K}$  be a convex set with bounded diameter  $D$ , and assume there exists a global minimizer for  $f$  in  $\mathcal{K}$ . Then Algorithm 2 with weights and learning rate as in Equation (1) ensures,*

$$f(\bar{y}_T) - \min_{x \in \mathbb{R}^d} f(x) \leq \mathcal{O}\left(\frac{DG + \beta D^2 \log(\beta D/G)}{T^2}\right)$$

**Remark:** Actually, in the smooth case we do not need a bound on the Lipschitz continuity, i.e.,  $G$  is only required in case that the objective is non-smooth. Concretely, if we know that  $f$  is smooth then we may use  $\eta_t = 2D \left(\sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2\right)^{-1/2}$ , which yields a rate of  $\mathcal{O}\left(\frac{\beta D^2 \log(\beta D/\|g_0\|)}{T^2}\right)$ .

Next we show that the exactly same algorithm provides guarantees in the general convex case,

**Theorem 3.2.** *Assume that  $f$  is convex and  $G$ -Lipschitz. Let  $\mathcal{K}$  be a convex set with bounded diameter  $D$ , and assume there exists a global minimizer for  $f$  in  $\mathcal{K}$ . Then Algorithm 2 with weights and learning rate as in Equation (1) ensures,*

$$f(\bar{y}_T) - \min_{x \in \mathbb{R}^d} f(x) \leq \mathcal{O}\left(GD\sqrt{\log T}/\sqrt{T}\right)$$

**Remark:** For non-smooth objectives, we can modify AcceleGrad and provide guarantees for the *constrained* setting. Concretely, using Alg. 2 with a projection step for the  $y_t$ 's, i.e.,  $y_{t+1} = \Pi_{\mathcal{K}}(x_{t+1} - \eta_t g_t)$ , then we can bound its error by  $f(\bar{y}_T) - \min_{x \in \mathcal{K}} f(x) \leq \mathcal{O}\left(GD\sqrt{\log T}/\sqrt{T}\right)$ . This holds even in the case where minimizer over  $\mathcal{K}$  is not a global one.

### 3.1 Analysis of the Smooth Case

Here we provide a proof sketch for Theorem 3.1. For brevity, we will use  $z \in \mathcal{K}$  to denote a *global minimizer* of  $f$  which belongs to  $\mathcal{K}$ .

Recall that Algorithm 2 outputs a weighted average of the queries. Consequently, we may employ Jensen's inequality to bound its error as follow,

$$f(\bar{y}_T) - f(z) \leq \frac{1}{\sum_{t=0}^{T-1} \alpha_t} \sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) . \quad (2)$$

Combining this with  $\sum_{t=0}^{T-1} \alpha_t \geq \Omega(T^2)$ , implies that in order to substantiate the proof it is sufficient to show that,  $\sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z))$ , is bounded by a constant. This is the bulk of the analysis.

We start with the following lemma which provides us with a bound on  $\alpha_t (f(y_{t+1}) - f(z))$ ,

**Lemma 3.1.** *Assume that  $f$  is convex and  $\beta$ -smooth. Then for any sequence of non-negative weights  $\{\alpha_t\}_{t \geq 0}$ , and learning rates  $\{\eta_t\}_{t \geq 0}$ , Algorithm 2 ensures the following to hold,*

$$\begin{aligned} \alpha_t(f(y_{t+1}) - f(z)) &\leq (\alpha_t^2 - \alpha_t)(f(y_t) - f(y_{t+1})) + \frac{\alpha_t^2}{2} \left( \beta - \frac{1}{\eta_t} \right) \|y_{t+1} - x_{t+1}\|^2 \\ &\quad + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2) \end{aligned}$$

Interestingly, choosing  $\eta_t \leq 1/\beta$ , implies that the above term,  $\frac{\alpha_t^2}{2} \left( \beta - \frac{1}{\eta_t} \right) \|y_{t+1} - x_{t+1}\|^2$ , does not contribute to the sum. We can show that this choice facilitates a concise analysis establishing an error of  $\mathcal{O}(\beta D^2/T^2)$  for  $\bar{y}_T^3$ .

Note however that our learning rate does not depend on  $\beta$ , and therefore the mentioned term is not necessarily negative. This issue is one of the main challenges in our analysis. Next we provide a proof sketch of Theorem 3.1. The full proof is deferred to Appendix A.

*Proof Sketch of Theorem 3.1.* Lemma 3.1 enables to decompose  $\sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z))$ ,

$$\begin{aligned} \sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) &\leq \underbrace{\sum_{t=0}^{T-1} \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2)}_{(A)} \\ &\quad + \underbrace{\sum_{t=0}^{T-1} (\alpha_t^2 - \alpha_t)(f(y_t) - f(y_{t+1}))}_{(B)} + \underbrace{\sum_{t=0}^{T-1} \frac{\alpha_t^2}{2} \left( \beta - \frac{1}{\eta_t} \right) \|y_{t+1} - x_{t+1}\|^2}_{(C)} \end{aligned} \quad (3)$$

Next we separately bound each of the above terms.

**(a) Bounding (A) :** Using the fact that  $\{1/\eta_t\}_{t \in [T]}$  is monotonically increasing allows to show,

$$\sum_{t=0}^{T-1} \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2) \leq \frac{1}{2} \sum_{t=1}^{T-1} \|z_t - z\|^2 \left( \frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right) + \frac{\|z_0 - z\|^2}{2\eta_0} \leq \frac{D^2}{2\eta_{T-1}} \quad (4)$$

where we used  $\|z_t - z\| \leq D$ .

**(b) Bounding (B) :** We will require the following property of the weights that we choose (Eq. (1)),

$$(\alpha_t^2 - \alpha_t) - (\alpha_{t-1}^2 - \alpha_{t-1}) \leq \alpha_{t-1}/2 \quad (5)$$

Now recall that  $z := \arg \min_{x \in \mathbb{R}^d} f(x)$ , and let us denote the sub-optimality of  $y_t$  by  $\delta_t$ , i.e.  $\delta_t = f(y_t) - f(z)$ . Noting that  $\delta_t \geq 0$  we may show the following,

$$\begin{aligned} \sum_{t=0}^{T-1} (\alpha_t^2 - \alpha_t) (f(y_t) - f(y_{t+1})) &= \sum_{t=0}^{T-1} (\alpha_t^2 - \alpha_t) (\delta_t - \delta_{t+1}) \\ &\leq \sum_{t=1}^{T-1} ((\alpha_t^2 - \alpha_t) - (\alpha_{t-1}^2 - \alpha_{t-1})) \delta_t \\ &\leq \frac{1}{2} \sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) \end{aligned} \quad (6)$$

Where the last inequality uses Equation (5) (see full proof for the complete derivation).

<sup>3</sup>While we do not spell out this analysis, it is a simplified version of our proof for Thm. 3.1.

**(c) Bounding (C) :** Let us denote  $\tau_* := \max \{t \in \{0, \dots, T-1\} : 2\beta \geq 1/\eta_t\}$ . We may now split the term (C) according to  $\tau_*$ ,

$$\begin{aligned}
(C) &= \sum_{t=0}^{\tau_*} \frac{\alpha_t^2}{2} \left( \beta - \frac{1}{\eta_t} \right) \|y_{t+1} - x_{t+1}\|^2 + \sum_{t=\tau_*+1}^{T-1} \frac{\alpha_t^2}{2} \left( \beta - \frac{1}{\eta_t} \right) \|y_{t+1} - x_{t+1}\|^2 \\
&\leq \frac{\beta}{2} \sum_{t=0}^{\tau_*} \alpha_t^2 \|y_{t+1} - x_{t+1}\|^2 - \frac{1}{4} \sum_{t=\tau_*+1}^{T-1} \frac{\alpha_t^2}{\eta_t} \|y_{t+1} - x_{t+1}\|^2 \\
&= \frac{\beta}{2} \sum_{t=0}^{\tau_*} \eta_t^2 \alpha_t^2 \|g_t\|^2 - \frac{1}{4} \sum_{t=\tau_*+1}^{T-1} \eta_t \alpha_t^2 \|g_t\|^2 \tag{7}
\end{aligned}$$

where in the second line we use  $2\beta \leq \frac{1}{\eta_t}$  which holds for  $t > \tau_*$ , implying that  $\beta - \frac{1}{\eta_t} \leq -\frac{1}{2\eta_t}$ ; in the last line we use  $\|y_{t+1} - x_{t+1}\| = \eta_t \|g_t\|$ .

**Final Bound :** Combining the bounds in Equations (4),(6),(7) into Eq. (3), and re-arranging gives,

$$\frac{1}{2} \sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) \leq \underbrace{\frac{D^2}{2\eta_{T-1}} - \frac{1}{4} \sum_{t=\tau_*+1}^{T-1} \eta_t \alpha_t^2 \|g_t\|^2}_{(*)} + \underbrace{\frac{\beta}{2} \sum_{t=0}^{\tau_*} \eta_t^2 \alpha_t^2 \|g_t\|^2}_{(**)} \tag{8}$$

We are now in the intricate part of the proof where we need to show that the above is bounded by a constant. As we show next this crucially depends on our choice of the learning rate. To simplify the proof sketch we assume to be using  $\eta_t = 2D \left( \sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2 \right)^{-1/2}$ , i.e. taking  $G = 0$  in the learning rate. We will require the following lemma before we go on,

**Lemma.** For any non-negative numbers  $a_1, \dots, a_n$  the following holds:

$$\sqrt{\sum_{i=1}^n a_i} \leq \sum_{i=1}^n \frac{a_i}{\sqrt{\sum_{j=1}^i a_j}} \leq 2\sqrt{\sum_{i=1}^n a_i}.$$

Equipped with the above lemma and using  $\eta_t$  explicitly enables to bound (\*),

$$\begin{aligned}
(*) &= \frac{D}{4} \left( \sum_{t=0}^{T-1} \alpha_t^2 \|g_t\|^2 \right)^{1/2} - \frac{D}{2} \sum_{t=\tau_*+1}^{T-1} \frac{\alpha_t^2 \|g_t\|^2}{\left( \sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2 \right)^{1/2}} \\
&\leq \frac{D}{4} \sum_{t=0}^{T-1} \frac{\alpha_t^2 \|g_t\|^2}{\left( \sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2 \right)^{1/2}} - \frac{D}{2} \sum_{t=\tau_*+1}^{T-1} \frac{\alpha_t^2 \|g_t\|^2}{\left( \sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2 \right)^{1/2}} \\
&\leq \frac{D}{4} \sum_{t=0}^{\tau_*} \frac{\alpha_t^2 \|g_t\|^2}{\left( \sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2 \right)^{1/2}} \\
&\leq \frac{D}{2} \left( \sum_{\tau=0}^{\tau_*} \alpha_\tau^2 \|g_\tau\|^2 \right)^{1/2} \\
&= \frac{D^2}{\eta_{\tau_*}} \leq 2\beta D^2
\end{aligned}$$

where in the last inequality we have used the definition of  $\tau_*$  which implies that  $1/\eta_{\tau_*} \leq 2\beta$ .

Using similar argumentation allows to bound the term (\*\*) by  $\mathcal{O}(\beta D^2 \log(\beta D / \|g_0\|))$ . Plugging these bounds back into Eq. (8) we get,

$$\sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) \leq \mathcal{O}(\beta D^2 \log(\beta D / \|g_0\|)).$$

Combining this with Eq. (2) and noting that  $\sum_{t=0}^{T-1} \alpha_t \geq T^2/32$ , concludes the proof.  $\square$

## 4 Stochastic Setting

This section discusses the stochastic optimization setup which is prevalent in Machine Learning scenarios. We formally describe this setup and prove that Algorithm 2, *without any modification*, is ensured to converge in this setting (Thm. 4.1). Conversely, the universal gradient methods presented in [27] rely on a line search procedure, which requires exact gradients and function values, and are therefore inappropriate for stochastic optimization.

As a related result we show that the AdaGrad algorithm (Alg. 1) is universal and is able to exploit small variance in order to ensure fast rates in the case of stochastic optimization with smooth expected loss (Thm. 4.2). We emphasize that AdaGrad does not require the smoothness nor a bound on the variance. Conversely, previous works with this type of guarantees, [35, 18], require the knowledge of *both of these parameters*.

**Setup:** We consider the problem of minimizing a convex function  $f : \mathbb{R}^d \mapsto \mathbb{R}$ . We assume that optimization lasts for  $T$  rounds; on each round  $t = 1, \dots, T$ , we may query a point  $x_t \in \mathbb{R}^d$ , and receive a *feedback*. After the last round, we choose  $\bar{x}_T \in \mathbb{R}^d$ , and our performance measure is the expected excess loss, defined as,

$$\mathbf{E}[f(\bar{x}_T)] - \min_{x \in \mathbb{R}^d} f(x).$$

Here we assume that our feedback is a first order noisy oracle such that upon querying this oracle with a point  $x$ , we receive a bounded and unbiased gradient estimate,  $\tilde{g}$ , such

$$\mathbf{E}[\tilde{g}|x] = \nabla f(x); \quad \& \quad \|\tilde{g}\| \leq G \tag{9}$$

We also assume that the internal coin tosses (randomizations) of the oracle are independent. It is well known that variants of Stochastic Gradient Descent (SGD) are ensured to output an estimate  $\bar{x}_T$  such that the excess loss is bounded by  $O(1/\sqrt{T})$  for the setups of stochastic convex optimization, [24]. Similarly to the offline setting we assume to be given a set  $\mathcal{K}$  with bounded diameter  $D$ , such that there exists a global optimum of  $f$  in  $\mathcal{K}$ .

The next theorem substantiates the guarantees of Algorithm 2 in the stochastic case,

**Theorem 4.1.** *Assume that  $f$  is convex and  $G$ -Lipschitz. Let  $\mathcal{K}$  be a convex set with bounded diameter  $D$ , and assume there exists a global minimizer for  $f$  in  $\mathcal{K}$ . Assume that we invoke Algorithm 2 but provide it with noisy gradient estimates (see Eq. (9)) rather than the exact ones. Then Algorithm 2 with weights and learning rate as in Equation (1) ensures,*

$$\mathbf{E}[f(\bar{y}_T)] - \min_{x \in \mathbb{R}^d} f(x) \leq \mathcal{O}\left(GD\sqrt{\log T}/\sqrt{T}\right)$$

The analysis of Theorem 4.1 goes along similar lines to the proof of its offline counterpart (Thm. 3.2).

It is well known that AdaGrad (Alg. 1) enjoys the standard rate of  $\mathcal{O}(GD/\sqrt{T})$  in the stochastic setting. The next lemma demonstrates that: **(i)** AdaGrad is universal, and **(ii)** AdaGrad implicitly make use of smoothness and small variance in the stochastic setting.

**Theorem 4.2.** *Assume that  $f$  is convex and  $\beta$ -smooth. Let  $\mathcal{K}$  be a convex set with bounded diameter  $D$ , and assume there exists a global minimizer for  $f$  in  $\mathcal{K}$ . Assume that we invoke AdaGrad (Alg. 1) but provide it with noisy gradient estimates (see Eq. (9)) rather than the exact ones. Then,*

$$\mathbf{E}[f(\bar{x}_T)] - \min_{x \in \mathbb{R}^d} f(x) \leq \mathcal{O}\left(\frac{\beta D^2}{T} + \frac{\sigma D}{\sqrt{T}}\right)$$

where  $\sigma^2$  is a bound on the variance of noisy gradients, i.e.,  $\forall x \in \mathbb{R}^d; \mathbf{E}[\|\tilde{g} - \nabla f(x)\|^2|x] \leq \sigma^2$ .

## 5 Experiments

In this section we compare AcceleGrad against AdaGrad (Alg. 1) and universal gradient methods [27], focusing on the effect of tuning parameters and the level of adaptivity.

We consider smooth ( $p = 2$ ) and non-smooth ( $p = 1$ ) regression problems of the form

$$\min_{x \in \mathbb{R}^d} F(x) := \|Ax - b\|_p^p.$$

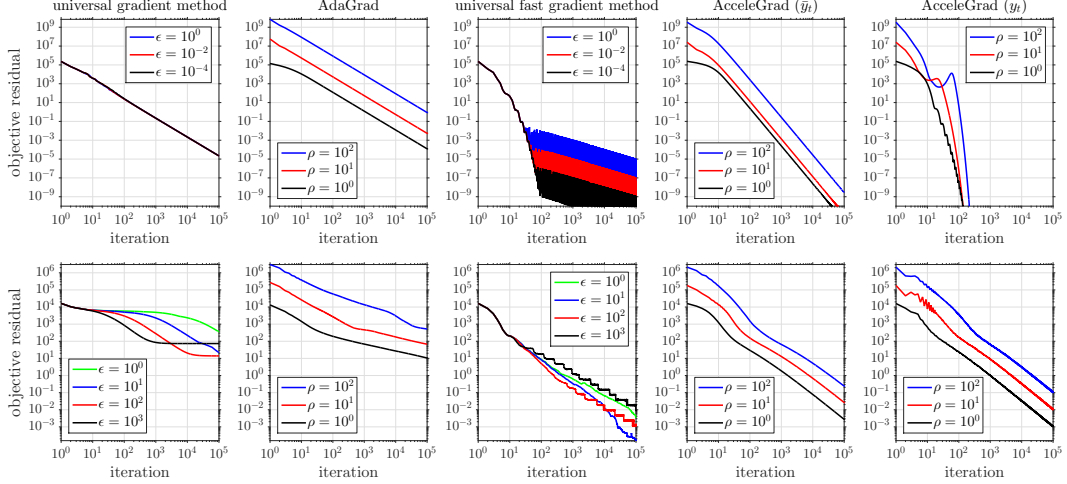


Figure 1: Comparison of universal methods at a smooth (*top*) and a non-smooth (*bottom*) problem.

We synthetically generate matrix  $A \in \mathbb{R}^{n \times d}$  and a point of interest  $x^\natural \in \mathbb{R}^d$  randomly, with entries independently drawn from standard Gaussian distribution. Then, we generate  $b = Ax^\natural + \omega$ , with Gaussian noise,  $w \sim \mathcal{N}(0, \sigma^2)$  and  $\sigma^2 = 10^{-2}$ . We fix  $n = 2000$  and  $d = 500$ .

Figure 1 presents the results for the offline optimization setting, where we provide the exact gradients of  $F$ . All methods are initialized at the origin, and we choose  $\mathcal{K}$  as the  $\ell_2$  norm ball of diameter  $D$ .

Universal gradient methods are based on an inexact line-search technique that requires an input parameter  $\epsilon$ . Moreover, these methods have convergence guarantees only up to  $\frac{\epsilon}{2}$ -suboptimality. For smooth problems, these methods perform better with smaller  $\epsilon$ . In stark contrast, for the non-smooth problems, small  $\epsilon$  causes late adaptation, and large  $\epsilon$  ends up with early saturation. Tuning is a major problem for these methods, since it requires rough knowledge of the optimal value.

Universal gradient method (also the fast version) provably requires two line-search iterations on average at each outer iteration. Consequently, it performs two data pass at each iteration (four for the fast version), while AdaGrad and AcceleGrad require only a single data pass.

The parameter  $\rho$  denotes the ratio between  $D/2$  and the distance between initial point and the solution. Parameter  $D$  plays a major role on the step-size of AdaGrad and AcceleGrad. Overestimating  $D$  causes an overshoot in the first iterations. AcceleGrad consistently overperforms AdaGrad in the deterministic setting. As a final note, it needs to be mentioned that the iterates  $y_t$  of AcceleGrad empirically converge faster than the averaged sequence  $\bar{y}_T$ . Note that for AcceleGrad we always take  $G = 0$ , i.e., use  $\eta_t = 2D \left( \sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2 \right)^{-1/2}$ .

We also study the stochastic setup (Appendix E), where we provide noisy gradients of  $F$  based on minibatches. As expected, universal line search methods *fail* in this case, while AcceleGrad converges and performs similarly to AdaGrad.

**Large batches:** In the appendix we show results on a real dataset which demonstrate the appeal of AcceleGrad in the large-minibatch regime. We show that with the increase of batch size the performance of AcceleGrad verses the number of gradient calculations does not degrade and might even *improve*. This is beneficial when we like to parallelize a stochastic optimization problem. Conversely, for AdaGrad we see a clear degradation of the performance as we increase the batch size.

## 6 Conclusion and Future Work

We have presented a novel universal method that may exploit smoothness in order to accelerate while still being able to successfully handle noisy feedback. Our current analysis only applies to unconstrained optimization problems. Extending our work to the constrained setting is a natural future direction. Another direction is to implicitly adapt the parameter  $D$ , this might be possible using ideas in the spirit of scale-free online algorithms, [29, 11].



## Acknowledgments

The authors would like to thank Zalán Borsos for his insightful comments on the manuscript.

This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement no 725594 - time-data). K.Y.L. is supported by the ETH Zurich Postdoctoral Fellowship and Marie Curie Actions for People COFUND program.

## References

- [1] Z. Allen-Zhu. Katyusha: The First Direct Acceleration of Stochastic Gradient Methods. In *STOC*, 2017. Full version available at <http://arxiv.org/abs/1603.05953>.
- [2] Z. Allen-Zhu and L. Orecchia. Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent. In *Proceedings of the 8th Innovations in Theoretical Computer Science*, ITCS '17, 2017. Full version available at <http://arxiv.org/abs/1407.1537>.
- [3] Y. Arjevani, S. Shalev-Shwartz, and O. Shamir. On lower and upper bounds in smooth and strongly convex optimization. *The Journal of Machine Learning Research*, 17(1):4303–4353, 2016.
- [4] H. Attouch and Z. Chbani. Fast inertial dynamics and fista algorithms in convex optimization. perturbation aspects. *arXiv preprint arXiv:1507.01367*, 2015.
- [5] J. Aujol and C. Dossal. Optimal rate of convergence of an ode associated to the fast gradient descent schemes for  $b > 0$ . 2017.
- [6] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.
- [7] S. Bubeck, Y. T. Lee, and M. Singh. A geometric alternative to nesterov’s accelerated gradient descent. *arXiv preprint arXiv:1506.08187*, 2015.
- [8] N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.
- [9] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.
- [10] M. B. Cohen, J. Diakonikolas, and L. Orecchia. On acceleration with noise-corrupted gradients. *arXiv preprint arXiv:1805.12591*, 2018.
- [11] A. Cutkosky and F. Orabona. Black-box reductions for parameter-free online learning in banach spaces. *arXiv preprint arXiv:1802.06293*, 2018.
- [12] J. Diakonikolas and L. Orecchia. Accelerated extra-gradient descent: A novel accelerated first-order method. *arXiv preprint arXiv:1706.04680*, 2017.
- [13] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.
- [14] N. Flammarion and F. Bach. From averaging to acceleration, there is only a step-size. In *Conference on Learning Theory*, pages 658–695, 2015.
- [15] S. Foucart and H. Rauhut. *A mathematical introduction to compressive sensing*, volume 1. Birkhäuser Basel, 2013.
- [16] R. Frostig, R. Ge, S. Kakade, and A. Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *International Conference on Machine Learning*, pages 2540–2548, 2015.
- [17] C. Hu, W. Pan, and J. T. Kwok. Accelerated gradient methods for stochastic optimization and online learning. In *Advances in Neural Information Processing Systems*, pages 781–789, 2009.
- [18] G. Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.
- [19] G. Lan. Bundle-level type methods uniformly optimal for smooth and nonsmooth convex optimization. *Mathematical Programming*, 149(1-2):1–45, 2015.

- [20] L. Lessard, B. Recht, and A. Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.
- [21] K. Levy. Online to offline conversions, universality and adaptive minibatch sizes. In *Advances in Neural Information Processing Systems*, pages 1612–1621, 2017.
- [22] H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pages 3384–3392, 2015.
- [23] H. B. McMahan and M. Streeter. Adaptive bound optimization for online convex optimization. *COLT 2010*, page 244, 2010.
- [24] A. Nemirovskii, D. B. Yudin, and E. Dawson. Problem complexity and method efficiency in optimization. 1983.
- [25] Y. Nesterov. A method of solving a convex programming problem with convergence rate  $o(1/k^2)$ . In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.
- [26] Y. Nesterov. Introductory lectures on convex optimization. 2004, 2003.
- [27] Y. Nesterov. Universal gradient methods for convex optimization problems. *Mathematical Programming*, 152(1-2):381–404, 2015.
- [28] A. Neumaier. Osga: a fast subgradient algorithm with optimal complexity. *Mathematical Programming*, 158(1-2):1–21, 2016.
- [29] F. Orabona and D. Pál. Scale-free algorithms for online linear optimization. In *International Conference on Algorithmic Learning Theory*, pages 287–301. Springer, 2015.
- [30] D. Scieur, A. d’Aspremont, and F. Bach. Regularized nonlinear acceleration. In *Advances In Neural Information Processing Systems*, pages 712–720, 2016.
- [31] S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *International Conference on Machine Learning*, pages 64–72, 2014.
- [32] W. Su, S. Boyd, and E. Candes. A differential equation for modeling nesterov’s accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems*, pages 2510–2518, 2014.
- [33] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.
- [34] A. Wibisono, A. C. Wilson, and M. I. Jordan. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016.
- [35] L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.
- [36] A. Yurtsever, Q. T. Dinh, and V. Cevher. A universal primal-dual convex optimization framework. In *Advances in Neural Information Processing Systems*, pages 3150–3158, 2015.

## A Proofs for the Smooth Case (Thm. 3.1)

Here we provide the complete proof of Theorem 3.1, and of the related lemmas. For brevity, we will use  $z \in \mathcal{K}$  to denote a *global minimizer* of  $f$  which belongs to  $\mathcal{K}$ .

Recall that Algorithm 2 outputs a weighted average of the queries. Consequently, we may employ Jensen's inequality to bound its error as follow,

$$f(\bar{y}_T) - f(z) \leq \frac{1}{\sum_{t=0}^{T-1} \alpha_t} \sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) . \quad (10)$$

Combining this with  $\sum_{t=0}^{T-1} \alpha_t \geq \Omega(T^2)$ , implies that in order to substantiate the proof it is sufficient to show that,  $\sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z))$ , is bounded by a constant. This is the bulk of the analysis.

We start by recalling Lemma 3.1 which provides us with a bound on  $\alpha_t (f(y_{t+1}) - f(z))$ ,

**Lemma** (Lemma 3.1). *Assume that  $f$  is convex and  $\beta$ -smooth. Then for any sequence of non-negative weights  $\{\alpha_t\}_{t \geq 0}$ , and learning rates  $\{\eta_t\}_{t \geq 0}$ , Algorithm 2 ensures the following to hold,*

$$\begin{aligned} \alpha_t (f(y_{t+1}) - f(z)) &\leq (\alpha_t^2 - \alpha_t) (f(y_t) - f(y_{t+1})) + \frac{\alpha_t^2}{2} \left( \beta - \frac{1}{\eta_t} \right) \|y_{t+1} - x_{t+1}\|^2 \\ &\quad + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2) \end{aligned}$$

The proof Lemma 3.1 is provided in Appendix A.1. We are now ready to prove Theorem 3.1.

*Proof of Theorem 3.1.* According to Lemma 3.1,

$$\begin{aligned} &\sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) \\ &\leq \underbrace{\sum_{t=0}^{T-1} \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2)}_{(A)} + \underbrace{\sum_{t=0}^{T-1} (\alpha_t^2 - \alpha_t) (f(y_t) - f(y_{t+1}))}_{(B)} \\ &\quad + \underbrace{\sum_{t=0}^{T-1} \frac{\alpha_t^2}{2} \left( \beta - \frac{1}{\eta_t} \right) \|y_{t+1} - x_{t+1}\|^2}_{(C)} \end{aligned} \quad (11)$$

It is natural to separately bound each of the sums above.

**(a) Bounding (A) :** Using the fact that  $\{1/\eta_t\}_{t \in [T]}$  is monotonically increasing we may bound (A) as follows,

$$\begin{aligned} \sum_{t=0}^{T-1} \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2) &\leq \frac{1}{2} \sum_{t=1}^{T-1} \|z_t - z\|^2 \left( \frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right) + \frac{\|z_0 - z\|^2}{2\eta_0} \\ &\leq \frac{D^2}{2\eta_{T-1}} \end{aligned} \quad (12)$$

where we used  $\|z_t - z\| \leq D$ .

**(b) Bounding (B) :** We will require the next lemma regarding the specific choice of the weights,

**Lemma A.1.** *The following holds for the  $\alpha_t$ 's which are described in Eq. (1),*

$$(\alpha_t^2 - \alpha_t) - (\alpha_{t-1}^2 - \alpha_{t-1}) \leq \alpha_{t-1}/2$$

Its proof appears in Appendix A.3.

We are now ready to bound (B). Recall that  $z := \arg \min_{x \in \mathbb{R}^d} f(x)$ , and let us denote the sub-optimality of  $y_t$  by  $\delta_t$ , i.e.  $\delta_t = f(y_t) - f(z)$ . Noting that  $\delta_t \geq 0$  we may show the following,

$$\begin{aligned}
& \sum_{t=0}^{T-1} (\alpha_t^2 - \alpha_t) (f(y_t) - f(y_{t+1})) \\
&= \sum_{t=0}^{T-1} (\alpha_t^2 - \alpha_t) (\delta_t - \delta_{t+1}) \\
&= \sum_{t=1}^{T-1} ((\alpha_t^2 - \alpha_t) - (\alpha_{t-1}^2 - \alpha_{t-1})) \delta_t + (\alpha_0^2 - \alpha_0) \delta_0 - (\alpha_{T-1}^2 - \alpha_{T-1}) \delta_T \\
&\leq \frac{1}{2} \sum_{t=1}^{T-1} \alpha_{t-1} \delta_t \\
&\leq \frac{1}{2} \sum_{t=1}^{T-1} \alpha_{t-1} \delta_t + \frac{1}{2} \alpha_{T-1} \delta_T \\
&= \frac{1}{2} \sum_{t=0}^{T-1} \alpha_t \delta_{t+1} \\
&= \frac{1}{2} \sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z))
\end{aligned} \tag{13}$$

where in the fourth line we use Lemma A.1, we also use  $\alpha_0^2 - \alpha_0 = 0$  and  $\alpha_{T-1}^2 - \alpha_{T-1} \geq 0$ .

**(c) Bounding (C) :** Let us denote  $\tau_*$  as follows,

$$\tau_* = \max \{t \in \{0, \dots, T-1\} : 2\beta \geq 1/\eta_t\} .$$

We may now split the last term as follows,

$$\begin{aligned}
& \sum_{t=0}^{T-1} \frac{\alpha_t^2}{2} \left( \beta - \frac{1}{\eta_t} \right) \|y_{t+1} - x_{t+1}\|^2 \\
&= \sum_{t=0}^{\tau_*} \frac{\alpha_t^2}{2} \left( \beta - \frac{1}{\eta_t} \right) \|y_{t+1} - x_{t+1}\|^2 + \sum_{t=\tau_*+1}^{T-1} \frac{\alpha_t^2}{2} \left( \beta - \frac{1}{\eta_t} \right) \|y_{t+1} - x_{t+1}\|^2 \\
&\leq \frac{\beta}{2} \sum_{t=0}^{\tau_*} \alpha_t^2 \|y_{t+1} - x_{t+1}\|^2 - \frac{1}{4} \sum_{t=\tau_*+1}^{T-1} \frac{\alpha_t^2}{\eta_t} \|y_{t+1} - x_{t+1}\|^2 \\
&= \frac{\beta}{2} \sum_{t=0}^{\tau_*} \eta_t^2 \alpha_t^2 \|g_t\|^2 - \frac{1}{4} \sum_{t=\tau_*+1}^{T-1} \eta_t \alpha_t^2 \|g_t\|^2
\end{aligned} \tag{14}$$

where in the third line we use  $2\beta \leq \frac{1}{\eta_t}$  which holds for  $t > \tau_*$ , implying that  $\beta - \frac{1}{\eta_t} \leq -\frac{1}{2\eta_t}$ ; in the fourth line we use  $\|y_{t+1} - x_{t+1}\| = \eta_t \|g_t\|$ .

**Final Bound :** Combining the bounds in Eq. (12)-(14) into Eq. (11), we obtain,

$$\begin{aligned}
\sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) &\leq \frac{D^2}{2\eta_{T-1}} + \frac{1}{2} \sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) \\
&\quad + \frac{\beta}{2} \sum_{t=0}^{\tau_*} \eta_t^2 \alpha_t^2 \|g_t\|^2 - \frac{1}{4} \sum_{t=\tau_*+1}^{T-1} \eta_t \alpha_t^2 \|g_t\|^2
\end{aligned}$$

Re-arranging we get,

$$\frac{1}{2} \sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) \leq \underbrace{\frac{D^2}{2\eta_{T-1}} - \frac{1}{4} \sum_{t=\tau_*+1}^{T-1} \eta_t \alpha_t^2 \|g_t\|^2}_{(*)} + \underbrace{\frac{\beta}{2} \sum_{t=0}^{\tau_*} \eta_t^2 \alpha_t^2 \|g_t\|^2}_{(**)} \quad (15)$$

This is the intricate part of the proof where we show that the above is bounded by a constant. This crucially depends on our choice of the learning rate, i.e.,  $\eta_t = 2D \left( G^2 + \sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2 \right)^{-1/2}$ . We require the following lemma (proof is found in Appendix A.4) before we go on,

**Lemma A.2.** *For any non-negative numbers  $a_1, \dots, a_n$  the following holds:*

$$\sqrt{\sum_{i=1}^n a_i} \leq \sum_{i=1}^n \frac{a_i}{\sqrt{\sum_{j=1}^i a_j}} \leq 2 \sqrt{\sum_{i=1}^n a_i}.$$

Equipped with the above lemma and using  $\eta_t$  explicitly enables to bound (\*),

$$\begin{aligned} (*) &= \frac{D}{4} \left( G^2 + \sum_{t=0}^{T-1} \alpha_t^2 \|g_t\|^2 \right)^{1/2} - \frac{D}{2} \sum_{t=\tau_*+1}^{T-1} \frac{\alpha_t^2 \|g_t\|^2}{\left( G^2 + \sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2 \right)^{1/2}} \\ &\leq \frac{D}{4} \left( \frac{G^2}{(G^2)^{1/2}} + \sum_{t=0}^{T-1} \frac{\alpha_t^2 \|g_t\|^2}{\left( G^2 + \sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2 \right)^{1/2}} \right) - \frac{D}{2} \sum_{t=\tau_*+1}^{T-1} \frac{\alpha_t^2 \|g_t\|^2}{\left( G^2 + \sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2 \right)^{1/2}} \\ &\leq \frac{DG}{4} + \frac{D}{4} \sum_{t=0}^{\tau_*} \frac{\alpha_t^2 \|g_t\|^2}{\left( G^2 + \sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2 \right)^{1/2}} \\ &\leq \frac{DG}{4} + \frac{D}{2} \left( \sum_{\tau=0}^{\tau_*} \alpha_\tau^2 \|g_\tau\|^2 \right)^{1/2} \\ &= \frac{DG}{4} + \frac{D^2}{\eta_{\tau_*}} \\ &\leq DG/4 + 2\beta D^2 \end{aligned} \quad (16)$$

where in the second line we use the left hand nequality of Lemma A.2; in the fourth line we use the right hand inequality of Lemma A.2 ; and in the last line we have used the definition of  $\tau_*$  which implies that  $1/\eta_{\tau_*} \leq 2\beta$ .

We will also require the following lemma (proof is found in Appendix A.5),

**Lemma A.3.** *For any non-negative real numbers  $a_1, \dots, a_n$ ,*

$$\sum_{i=1}^n \frac{a_i}{1 + \sum_{j=1}^i a_j} \leq 1 + \log \left( 1 + \sum_{i=1}^n a_i \right).$$

Equipped with the above lemma and using  $\eta_t$  explicitly enables to bound (\*\*),

$$\begin{aligned}
\frac{\beta}{2} \sum_{t=0}^{\tau_*} \eta_t^2 \alpha_t^2 \|g_t\|^2 &= \frac{4\beta D^2}{2} \sum_{t=0}^{\tau_*} \frac{\alpha_t^2 \|g_t\|^2}{G^2 + \sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2} \\
&= 2\beta D^2 \sum_{t=0}^{\tau_*} \frac{\alpha_t^2 (\|g_t\|/G)^2}{1 + \sum_{\tau=0}^t \alpha_\tau^2 (\|g_\tau\|/G)^2} \\
&\leq 2\beta D^2 \left( 1 + \log \left( (G/G)^2 + \sum_{\tau=0}^{\tau_*} \alpha_\tau^2 (\|g_\tau\|/G)^2 \right) \right) \\
&= 2\beta D^2 \left( 1 + \log \left( \frac{4D^2/G^2}{\eta_{\tau_*}^2} \right) \right) \\
&\leq 2\beta D^2 (1 + 2 \log(4\beta D/G)) \tag{17}
\end{aligned}$$

where in the third line we used Lemma A.3, and in the last line we have used the definition of  $\tau_*$  which implies that  $1/\eta_{\tau_*} \leq 2\beta$ . Combining Equations (16), (17) back into Eq. (15) and using Jensen's inequality we are now ready to establish the final bound,

$$\begin{aligned}
f(\bar{y}_T) - f(z) &\leq \frac{\sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z))}{\sum_{t=0}^{T-1} \alpha_t} \\
&\leq \frac{DG/2 + 8\beta D^2 (1 + \log(4\beta D/G))}{T^2/32} \\
&= O\left(\frac{DG + \beta D^2 \log(\beta D/G)}{T^2}\right).
\end{aligned}$$

where we have used  $\alpha_t \geq \frac{1}{4}(t+1)$  and therefore  $\sum_{t=0}^{T-1} \alpha_t \geq T^2/32$ . □

### A.1 Proof of Lemma 3.1

*Proof.* Our starting point is bounding  $\alpha_t(f(x_{t+1}) - f(z))$  which can be decomposed as follows,

$$\begin{aligned}
\alpha_t(f(x_{t+1}) - f(z)) &\leq \alpha_t g_t \cdot (x_{t+1} - z) \\
&= \alpha_t g_t \cdot (z_t - z) + \alpha_t g_t \cdot (x_{t+1} - z_t) \tag{18}
\end{aligned}$$

where we use  $g_t = \nabla f(x_{t+1})$  in conjunction with the gradient inequality. Let us now bound the terms in the above equation.

**(a) Bounding  $\alpha_t g_t \cdot (z_t - z)$ :** The next lemma enables to bound this term,

**Lemma A.4.** *The following holds,*

$$\alpha_t g_t \cdot (z_t - z) \leq \left( \alpha_t g_t \cdot (z_t - z_{t+1}) - \frac{1}{2\eta_t} \|z_t - z_{t+1}\|^2 \right) + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2)$$

The proof of Lemma A.4 is provided in Appendix A.2.

We can now relate the first term in the above lemma to  $y_{t+1}$ . Define  $v = \tau_t z_{t+1} + (1 - \tau_t)y_t \in \mathcal{K}$ , and notice that  $x_{t+1} - v = \tau_t(z_t - z_{t+1})$ . Using this we may write,

$$\begin{aligned}
& \alpha_t g_t \cdot (z_t - z_{t+1}) - \frac{1}{2\eta_t} \|z_t - z_{t+1}\|^2 \\
&= \frac{\alpha_t}{\tau_t} g_t \cdot (x_{t+1} - v) - \frac{1}{2\eta_t \tau_t^2} \|x_{t+1} - v\|^2 \\
&= \alpha_t^2 \left( g_t \cdot (x_{t+1} - v) - \frac{1}{2\eta_t} \|x_{t+1} - v\|^2 \right) \\
&= \alpha_t^2 g_t \cdot x_{t+1} - \alpha_t^2 \left( g_t \cdot v + \frac{1}{2\eta_t} \|x_{t+1} - v\|^2 \right) \\
&\leq \alpha_t^2 g_t \cdot x_{t+1} - \alpha_t^2 \left( g_t \cdot y_{t+1} + \frac{1}{2\eta_t} \|x_{t+1} - y_{t+1}\|^2 \right) \\
&= \alpha_t^2 g_t \cdot (x_{t+1} - y_{t+1}) - \frac{\alpha_t^2}{2\eta_t} \|x_{t+1} - y_{t+1}\|^2
\end{aligned} \tag{19}$$

where we use  $\tau_t = 1/\alpha_t$ ; also in the inequality we use the following equivalent form for the update rule of  $y_{t+1}$ ,

$$y_{t+1} = \arg \min_{x \in \mathbb{R}^d} g_t \cdot x + \frac{1}{2\eta_t} \|x - x_{t+1}\|^2 .$$

this equivalence can be directly validated by finding the global optimum of the above objective and showing that it is obtained by choosing  $y_{t+1} = x_{t+1} - \eta_t g_t$ .

Combining Eq. (19) with Lemma A.4 gives,

$$\alpha_t g_t \cdot (z_t - z) \leq \alpha_t^2 g_t \cdot (x_{t+1} - y_{t+1}) - \frac{\alpha_t^2}{2\eta_t} \|x_{t+1} - y_{t+1}\|^2 + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2) \tag{20}$$

**(b) Bounding  $\alpha_t g_t \cdot (x_{t+1} - z_t)$ :** Notice that re-arranging the relation between  $x_{t+1}, y_t, z_t$  (recall  $x_{t+1} = \tau_t z_t + (1 - \tau_t)y_t$ ) gives,

$$x_{t+1} - z_t = r_t (y_t - x_{t+1}) \tag{21}$$

where we denote  $r_t = (1 - \tau_t)/\tau_t$ . Also note that the smoothness of  $f$  implies,

$$f(y_{t+1}) - f(x_{t+1}) \leq g_t \cdot (y_{t+1} - x_{t+1}) + \frac{\beta}{2} \|y_{t+1} - x_{t+1}\|^2 \tag{22}$$

Combining Eq. (21) and (22) we get,

$$\begin{aligned}
& g_t \cdot (x_{t+1} - z_t) \\
&= r_t \nabla f(x_{t+1}) \cdot (y_t - x_{t+1}) \\
&\leq r_t (f(y_t) - f(x_{t+1})) \\
&= r_t (f(y_t) - f(y_{t+1})) + (r_t + 1) (f(y_{t+1}) - f(x_{t+1})) - (f(y_{t+1}) - f(x_{t+1})) \\
&\leq (\alpha_t - 1) (f(y_t) - f(y_{t+1})) + \alpha_t \left( g_t \cdot (y_{t+1} - x_{t+1}) + \frac{\beta}{2} \|y_{t+1} - x_{t+1}\|^2 \right) \\
&\quad - (f(y_{t+1}) - f(x_{t+1}))
\end{aligned} \tag{23}$$

where second line uses the gradient inequality. We have also used  $r_t = (1 - \tau_t)/\tau_t = \alpha_t - 1$  (see Alg. 2).

**(c) Bounding  $\alpha_t \cdot (f(y_{t+1}) - f(z))$ :** Combining Equations (18), (20) and (23) we get,

$$\begin{aligned}
& \alpha_t(f(x_{t+1}) - f(z)) \\
& \leq \alpha_t g_t \cdot (z_t - z) + \alpha_t g_t \cdot (x_{t+1} - z_t) \\
& \leq \left\{ \alpha_t^2 g_t \cdot (x_{t+1} - y_{t+1}) - \frac{\alpha_t^2}{2\eta_t} \|x_{t+1} - y_{t+1}\|^2 + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2) \right\} \\
& \quad + (\alpha_t^2 - \alpha_t)(f(y_t) - f(y_{t+1})) + \alpha_t^2 \left( g_t \cdot (y_{t+1} - x_{t+1}) + \frac{\beta}{2} \|y_{t+1} - x_{t+1}\|^2 \right) \\
& \quad - \alpha_t (f(y_{t+1}) - f(x_{t+1})) \\
& = (\alpha_t^2 - \alpha_t)(f(y_t) - f(y_{t+1})) + \frac{\alpha_t^2}{2} \left( \beta - \frac{1}{\eta_t} \right) \|y_{t+1} - x_{t+1}\|^2 \\
& \quad + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2) - \alpha_t (f(y_{t+1}) - f(x_{t+1}))
\end{aligned}$$

Re-arranging the above equation implies,

$$\begin{aligned}
& \alpha_t(f(y_{t+1}) - f(z)) \\
& \leq (\alpha_t^2 - \alpha_t)(f(y_t) - f(y_{t+1})) + \frac{\alpha_t^2}{2} \left( \beta - \frac{1}{\eta_t} \right) \|y_{t+1} - x_{t+1}\|^2 \\
& \quad + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2)
\end{aligned}$$

which concludes the proof.  $\square$

## A.2 Proof of Lemma A.4

*Proof.* Writing the update of the  $z_t$ 's explicitly we have,

$$z_{t+1} \leftarrow \arg \min_{x \in \mathcal{K}} \|x - (z_t - \eta_t \alpha_t g_t)\|^2.$$

Simplifying the above implies the following equivalent form,

$$z_{t+1} \leftarrow \arg \min_{x \in \mathcal{K}} \alpha_t g_t \cdot x + \frac{1}{\eta_t} \mathcal{R}_{z_t}(x),$$

where  $\mathcal{R}_{z_t}(x) := \|x - z_t\|^2/2$ . Since  $z_{t+1}$  is a solution of the above minimization problem it satisfies the first order optimality conditions, i.e.  $\forall z \in \mathcal{K}$ ,

$$\alpha_t g_t \cdot (z - z_{t+1}) + \frac{1}{\eta_t} \nabla \mathcal{R}_{z_t}(z_{t+1}) \cdot (z - z_{t+1}) \geq 0 \quad (24)$$

which follows by the first order optimality conditions for  $z_{t+1}$ . We are now ready to complete the proof,

$$\begin{aligned}
& \alpha_t g_t \cdot (z_t - z) = \alpha_t g_t \cdot (z_t - z_{t+1}) + \alpha_t g_t \cdot (z_{t+1} - z) \\
& \leq \alpha_t g_t \cdot (z_t - z_{t+1}) - \frac{1}{\eta_t} \nabla \mathcal{R}_{z_t}(z_{t+1}) \cdot (z_{t+1} - z) \\
& = \alpha_t g_t \cdot (z_t - z_{t+1}) - \frac{1}{2\eta_t} \|z_t - z_{t+1}\|^2 + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2)
\end{aligned}$$

where the second line follows due to Eq. (24), and the second line is due to following lemma (which may be easily extended to general Bergman divergences),

**Lemma A.5.** *Let  $u, v, z \in \mathbb{R}^d$ , and let  $\mathcal{R}_v(x) := \frac{1}{2} \|x - v\|^2$ , then*

$$-\nabla \mathcal{R}_v(u) \cdot (u - z) = \frac{1}{2} \|v - z\|^2 - \frac{1}{2} \|u - z\|^2 - \frac{1}{2} \|u - v\|^2$$

Below we provide the proof of this lemma.  $\square$



### A.2.1 Proof of Lemma A.5

*Proof.* Noticing that  $-\nabla\mathcal{R}_v(u) = v - u$  the lemma may be validated by a direct calculation. Indeed,  $-\nabla\mathcal{R}_v(u) \cdot (u - z) = -v \cdot z + u \cdot z + u \cdot v - \|u\|^2$ . Also,

$$\|v - z\|^2 - \|u - z\|^2 - \|u - v\|^2 = -2v \cdot z + 2u \cdot z + 2u \cdot v - 2\|u\|^2$$

□

### A.3 Proof of Lemma A.1

*Proof.* For  $t \leq 3$  we have  $\alpha_t^2 - \alpha_t = 0$  and the lemma immediately follows. For  $t > 3$  we have,

$$(\alpha_t^2 - \alpha_t) - (\alpha_{t-1}^2 - \alpha_{t-1}) = \frac{(t+1)^2 - 4(t+1)}{16} - \frac{t^2 - 4t}{16} = \frac{2t-3}{8} \leq \alpha_{t-1}/2$$

□

### A.4 Proof of Lemma A.2

*Proof. First direction:* We will prove this part by induction. The base case,  $n = 1$ , immediately holds. For the induction step assume that the lemma holds for  $n - 1$  and let us show it holds for  $n$ . By the induction assumption,

$$\sum_{i=1}^n \frac{a_i}{\sqrt{\sum_{j=1}^i a_j}} \geq \sqrt{\sum_{i=1}^{n-1} a_i} + \frac{a_n}{\sqrt{\sum_{i=1}^n a_i}} = \sqrt{Z-x} + \frac{x}{\sqrt{Z}}$$

where we denote  $x := a_n$  and  $Z = \sum_{i=1}^n a_i$  (note that  $x \leq Z$ ). Thus, in order to prove the lemma it is sufficient to show that,

$$\sqrt{Z-x} + \frac{x}{\sqrt{Z}} \geq \sqrt{Z},$$

which we do next. Multiplying both sides by  $\sqrt{Z}$  we get that the above is equivalent to,

$$\sqrt{Z^2 - xZ} \geq Z - x$$

Taking the square of the above and re-ordering we get that the above is equivalent to,

$$x \leq Z$$

Which holds in our case since  $x = a_n \leq \sum_{i=1}^n a_i = Z$ . This concludes the first part of the proof.

**Second direction:** The second inequality in the lemma is due to Lemma 7 in [23]. For completeness we include their proof.

This part is also proved by induction. The base case,  $n = 1$ , immediately holds. For the induction step assume that the lemma holds for  $n - 1$  and let us show it holds for  $n$ . By the induction assumption,

$$\sum_{i=1}^n \frac{a_i}{\sqrt{\sum_{j=1}^i a_j}} \leq 2\sqrt{\sum_{i=1}^{n-1} a_i} + \frac{a_n}{\sqrt{\sum_{i=1}^n a_i}} = 2\sqrt{Z-x} + \frac{x}{\sqrt{Z}}$$

where we denote  $x := a_n$  and  $Z = \sum_{i=1}^n a_i$  (note that  $x \leq Z$ ). The derivative of the right hand side with respect to  $x$  is  $-\frac{1}{\sqrt{Z-x}} + \frac{1}{\sqrt{Z}}$ , which is negative for  $x \geq 0$ . Thus, subject to the constraint  $x \geq 0$ , the right hand side is maximized at  $x = 0$ , and is therefore at most  $2\sqrt{Z}$ . This concludes the second part of the proof. □

### A.5 Proof of Lemma A.3

*Proof.* We will prove the statement by induction over  $n$ . The base case  $n = 1$  holds since,

$$\frac{a_1}{1 + a_1} \leq 1 \leq 1 + \log(1 + a_1).$$

For the induction step, let us assume that the guarantee holds for  $n - 1$ , which implies that for any  $a_1, \dots, a_n \geq 0$ ,

$$\sum_{i=1}^n \frac{a_i}{1 + \sum_{j=1}^i a_j} \leq 1 + \log\left(1 + \sum_{i=1}^{n-1} a_i\right) + \frac{a_n}{1 + \sum_{i=1}^n a_i}.$$

The above suggests that establishing following inequality concludes the proof,

$$1 + \log\left(1 + \sum_{i=1}^{n-1} a_i\right) + \frac{a_n}{1 + \sum_{i=1}^n a_i} \leq 1 + \log\left(1 + \sum_{i=1}^n a_i\right). \quad (25)$$

Using the notation  $x = a_n / (1 + \sum_{i=1}^{n-1} a_i)$ , Equation (25) is equivalent to the following,

$$\log(x + 1) - \frac{x}{1 + x} \geq 0.$$

However, it is immediate to validate that the function  $M(x) = \log(x + 1) - \frac{x}{1+x}$ , is non-negative for any  $x \geq 0$ , which establishes the lemma.  $\square$

## B Proofs for the General Convex Case (Thm. 3.2)

Here we provide the complete proof of Theorem 3.2, and of the related lemmas. For brevity, we will use  $z \in \mathcal{K}$  to denote a *global minimizer* of  $f$  which belongs to  $\mathcal{K}$ .

Recall that Algorithm 2 outputs a weighted average of the queries. Consequently, we may employ Jensen's inequality to bound its error as follow,

$$f(\bar{y}_T) - f(z) \leq \frac{1}{\sum_{t=0}^{T-1} \alpha_t} \sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) . \quad (26)$$

Combining this with  $\sum_{t=0}^{T-1} \alpha_t \geq \Omega(T^2)$ , implies that in order to substantiate the proof it is sufficient to show that,  $\sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z))$ , is bounded by  $\tilde{\mathcal{O}}(T^{3/2})$ . This is the bulk of the analysis.

We start with the following lemma which provides us with a bound on  $\alpha_t (f(y_{t+1}) - f(z))$ ,

**Lemma B.1.** *Assume that  $f$  is convex and  $G$ -Lipschitz. Then for any sequence of non-negative weights  $\{\alpha_t\}_{t \geq 0}$ , and learning rates  $\{\eta_t\}_{t \geq 0}$ , Algorithm 2 ensures the following to hold,*

$$\begin{aligned} & \alpha_t (f(y_{t+1}) - f(z)) \\ & \leq \eta_t \alpha_t^2 \|g_t\|^2 + \eta_t \alpha_t^2 \|g_t\| G + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2) + (\alpha_t^2 - \alpha_t) (f(y_t) - f(y_{t+1})) \end{aligned}$$

The proof of Lemma B.1 is provided in Appendix B.1. We are now ready to prove Theorem 3.2.

*Proof of Theorem 3.2.* According to Lemma B.1,

$$\begin{aligned} & \sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) \\ & \leq \underbrace{\sum_{t=0}^{T-1} \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2)}_{(A)} + \underbrace{\sum_{t=0}^{T-1} (\alpha_t^2 - \alpha_t) (f(y_t) - f(y_{t+1}))}_{(B)} \\ & \quad + \underbrace{\sum_{t=0}^{T-1} \eta_t \alpha_t^2 \|g_t\|^2}_{(C)} + \underbrace{\sum_{t=0}^{T-1} \eta_t \alpha_t^2 \|g_t\| G}_{(D)} \end{aligned} \quad (27)$$

It is natural to separately bound each of the sums above.

**(a) Bounding (A) :** Similarly to part (a) in the proof of Theorem 3.1 we can show the following to hold,

$$\sum_{t=0}^{T-1} \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2) \leq \frac{D^2}{\eta_{T-1}} \quad (28)$$

**(b) Bounding (B) :** Similarly to part (b) in the proof of Theorem 3.1 we can show the following to hold for  $z = \arg \min_{z \in \mathbb{R}^d} f(x)$ ,

$$\sum_{t=0}^{T-1} (\alpha_t^2 - \alpha_t) (f(y_t) - f(y_{t+1})) \leq \frac{1}{2} \sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) \quad (29)$$

**(c) Bounding (C) :** Note that by the definition of  $\eta_t$  we have

$$\eta_t = \frac{2D}{\left(G^2 + \sum_{\tau=1}^t \alpha_\tau^2 \|g_\tau\|^2\right)^{1/2}} \leq \frac{2D}{\left(\sum_{\tau=1}^t \alpha_\tau^2 \|g_\tau\|^2\right)^{1/2}} .$$

Using the above inequality we get,

$$\sum_{t=0}^{T-1} \eta_t \alpha_t^2 \|g_t\|^2 \leq 2D \sum_{t=0}^{T-1} \frac{\alpha_t^2 \|g_t\|^2}{\left(\sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2\right)^{1/2}} \leq 4D \sqrt{\sum_{t=0}^{T-1} \alpha_t^2 \|g_t\|^2} \quad (30)$$

where the second inequality uses Lemma A.2.

**(d) Bounding (D) :** Writing down  $\eta_t$  explicitly we get,

$$\begin{aligned} \sum_{t=0}^{T-1} \eta_t \alpha_t^2 \|g_t\| G &= 2DG \sum_{t=0}^{T-1} \frac{\alpha_t^2 \|g_t\|}{\left(G^2 + \sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2\right)^{1/2}} \\ &\leq 2DGT \sum_{t=0}^{T-1} \frac{\alpha_t \|g_t\|}{\left(G^2 + \sum_{\tau=0}^t \alpha_\tau^2 \|g_\tau\|^2\right)^{1/2}} \\ &= 2DGT \sum_{t=0}^{T-1} \frac{\alpha_t (\|g_t\|/G)}{\left(1 + \sum_{\tau=0}^t \alpha_\tau^2 (\|g_\tau\|/G)^2\right)^{1/2}} \\ &\leq 10DG \sqrt{\log T} \cdot T^{3/2}. \end{aligned} \quad (31)$$

where we used  $\forall t \leq T; \alpha_t \leq T$ . The last line uses the following lemma (see proof in Appendix B.2),

**Lemma B.2.** Consider the  $\alpha_t$ 's used by our algorithm, i.e.,

$$\alpha_t = \begin{cases} 1 & 0 \leq t \leq 2 \\ \frac{1}{4}(t+1) & t \geq 3 \end{cases}$$

And assume a sequence of non-negative numbers,  $b_0, b_1, \dots, b_{T-1} \in [0, 1]$ . Then the following holds,

$$\sum_{t=0}^{T-1} \frac{\alpha_t b_t}{\left(1 + \sum_{\tau=0}^t \alpha_\tau^2 b_\tau^2\right)^{1/2}} \leq 5\sqrt{\log T} \sqrt{T}$$

**Final Bound :** Combining the bounds on the different terms, Eq. (28)-(31), together with Eq. (27), we have,

$$\begin{aligned} &\sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) \\ &\leq \frac{D^2}{\eta_{T-1}} + \frac{1}{2} \sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) \\ &\quad + 4D \sqrt{\sum_{t=0}^{T-1} \alpha_t^2 \|g_t\|^2} + 10DG \sqrt{\log T} \cdot T^{3/2} \end{aligned}$$

Re-arranging and using the explicit expression for  $\eta_{T-1}$  we get,

$$\begin{aligned} &\frac{1}{2} \sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) \\ &\leq 5D \sqrt{G^2 + \sum_{t=0}^{T-1} \alpha_t^2 \|g_t\|^2} + 10DG \sqrt{\log T} \cdot T^{3/2} \\ &\leq 5DG \sqrt{1 + T^3} + 10DG \sqrt{\log T} \cdot T^{3/2} \\ &\leq 20DG \sqrt{\log T} \cdot T^{3/2}. \end{aligned}$$

where we have used  $\|g_t\| \leq G$ , and also,  $\alpha_t \leq t + 1$  implying that  $\sum_{t=0}^{T-1} \alpha_t^2 \leq T^3$ .

Using Jensen's inequality we are now ready to establish the final bound,

$$\begin{aligned} f(\bar{y}_T) - f(z) &\leq \frac{\sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z))}{\sum_{t=0}^{T-1} \alpha_t} \\ &\leq \frac{40 \cdot DG \sqrt{\log T} \cdot T^{3/2}}{T^2/32} \\ &= O\left(DG \sqrt{\log T} / \sqrt{T}\right) \end{aligned}$$

where we have used  $\alpha_t \geq \frac{1}{4}(t+1)$  and therefore  $\sum_{t=0}^{T-1} \alpha_t \geq T^2/32$ . □

### B.1 Proof of Lemma B.1

*Proof.* Our starting point is bounding  $\alpha_t(f(x_{t+1}) - f(z))$  which can be decomposed as follows,

$$\begin{aligned} \alpha_t(f(x_{t+1}) - f(z)) &\leq \alpha_t g_t \cdot (x_{t+1} - z) \\ &= \alpha_t g_t \cdot (z_t - z) + \alpha_t g_t \cdot (x_{t+1} - z_t) \end{aligned} \quad (32)$$

where we use  $g_t = \nabla f(x_{t+1})$  in conjunction with the gradient inequality. Let us now bound the terms in the above equation.

**(a) Bounding  $\alpha_t g_t \cdot (z_t - z)$ :** Similarly to the proof of Lemma 3.1 we can show the following to hold (see Eq. (20) in Lemma 3.1),

$$\alpha_t g_t \cdot (z_t - z) \leq \alpha_t^2 g_t \cdot (x_{t+1} - y_{t+1}) - \frac{\alpha_t^2}{2\eta_t} \|x_{t+1} - y_{t+1}\|^2 + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2)$$

Combining the above with  $\|x_{t+1} - y_{t+1}\| = \eta_t \|g_t\|$  implies,

$$\alpha_t g_t \cdot (z_t - z) \leq \eta_t \alpha_t^2 \|g_t\|^2 + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2) \quad (33)$$

**(b) Bounding  $\alpha_t g_t \cdot (x_{t+1} - z_t)$ :** Notice that re-arranging the relation between  $x_{t+1}, y_t, z_t$  (recall  $x_{t+1} = \tau_t z_t + (1 - \tau_t) y_t$ ) gives,

$$x_{t+1} - z_t = r_t (y_t - x_{t+1})$$

where we denote  $r_t = (1 - \tau_t)/\tau_t$ . Using the above we get,

$$\begin{aligned} &g_t \cdot (x_{t+1} - z_t) \\ &= r_t \nabla f(x_{t+1}) \cdot (y_t - x_{t+1}) \\ &\leq (\alpha_t - 1) (f(y_t) - f(x_{t+1})) \\ &\leq \alpha_t (f(y_{t+1}) - f(x_{t+1})) - (f(y_{t+1}) - f(x_{t+1})) + (\alpha_t - 1) (f(y_t) - f(y_{t+1})) \\ &\leq \alpha_t G \eta_t \|g_t\| - (f(y_{t+1}) - f(x_{t+1})) + (\alpha_t - 1) (f(y_t) - f(y_{t+1})) \end{aligned} \quad (34)$$

where second line uses the gradient inequality, in the third line we used  $r_t = (1 - \tau_t)/\tau_t = \alpha_t - 1$  (see Alg. 2); and in the last line we used  $|f(y_{t+1}) - f(x_{t+1})| \leq G \|y_{t+1} - x_{t+1}\| \leq G \eta_t \|g_t\|$ , which follows by the  $G$ -Lipschitzness of  $f$ .

**(c) Bounding  $\alpha_t \cdot (f(y_{t+1}) - f(z))$ :** Combining Equations (32), (33), (34) we get,

$$\begin{aligned} &\alpha_t (f(x_{t+1}) - f(z)) \\ &\leq \alpha_t g_t \cdot (z_t - z) + \alpha_t g_t \cdot (x_{t+1} - z_t) \\ &\leq \left\{ \eta_t \alpha_t^2 \|g_t\|^2 + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2) \right\} \\ &\quad + (\alpha_t^2 - \alpha_t) (f(y_t) - f(y_{t+1})) + \eta_t \alpha_t^2 \|g_t\| G - \alpha_t (f(y_{t+1}) - f(x_{t+1})) \end{aligned}$$

Re-arranging the above equation and we get,

$$\begin{aligned} & \alpha_t(f(y_{t+1}) - f(z)) \\ & \leq \eta_t \alpha_t^2 \|g_t\|^2 + \eta_t \alpha_t^2 \|g_t\| G + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2) + (\alpha_t^2 - \alpha_t) (f(y_t) - f(y_{t+1})) \end{aligned}$$

which concludes the proof.  $\square$

## B.2 Proof of Lemma B.2

*Proof.* Let us define the following time variables,

$$T_0 = \max \left\{ t \in \{0, \dots, T-1\} : \sum_{\tau=0}^t \alpha_\tau^2 b_\tau^2 \leq 1 \right\}$$

and for any  $k \geq 1$

$$T_k = \max \left\{ t \in \{0, \dots, T-1\} : 4^{k-1} < \sum_{\tau=0}^t \alpha_\tau^2 b_\tau^2 \leq 4^k \right\}$$

By the definition of  $T_0$ , the following applies,

$$\sum_{\tau=0}^{T_0} \alpha_\tau b_\tau \leq \sqrt{T_0 + 1} \left( \sum_{\tau=0}^{T_0} \alpha_\tau^2 b_\tau^2 \right)^{1/2} \leq \sqrt{T}. \quad (35)$$

where in the first inequality we use  $\|u\|_1 \leq \sqrt{n}\|u\|_2$ ,  $\forall u \in \mathbb{R}^n$ , in the second inequality we use the definition of  $T_0$  together with  $T_0 \leq T-1$ .

For the other time variables we can similarly show the following bounds, i.e.,  $\forall k \geq 1$ ,

$$\sum_{\tau=T_{k-1}+1}^{T_k} \alpha_\tau b_\tau \leq \sqrt{T_k - T_{k-1}} \left( \sum_{\tau=T_{k-1}+1}^{T_k} \alpha_\tau^2 b_\tau^2 \right)^{1/2} \leq \sqrt{T_k - T_{k-1}} \cdot 2^k \quad (36)$$

where in the first inequality we use  $\|u\|_1 \leq \sqrt{n}\|u\|_2$ ,  $\forall u \in \mathbb{R}^n$ , in the second inequality we use the definition of  $T_k$ .

Using the definition of the time variables together with Equations (35),(36) we get,

$$\begin{aligned} & \sum_{t=0}^{T-1} \frac{\alpha_t b_t}{\left(1 + \sum_{\tau=0}^t \alpha_\tau^2 b_\tau^2\right)^{1/2}} \\ & = \sum_{t=0}^{T_0} \frac{\alpha_t b_t}{\left(1 + \sum_{\tau=0}^t \alpha_\tau^2 b_\tau^2\right)^{1/2}} + \sum_{k \geq 1} \sum_{t=T_{k-1}+1}^{T_k} \frac{\alpha_t b_t}{\left(1 + \sum_{\tau=0}^t \alpha_\tau^2 b_\tau^2\right)^{1/2}} \\ & \leq \sum_{t=0}^{T_0} \alpha_t b_t + \sum_{k \geq 1} \sum_{t=T_{k-1}+1}^{T_k} \frac{\alpha_t b_t}{(1 + 4^{k-1})^{1/2}} \\ & \leq \sqrt{T} + \sum_{k \geq 1} \frac{1}{2^{k-1}} \sum_{t=T_{k-1}+1}^{T_k} \alpha_t b_t \\ & \leq \sqrt{T} + 2 \sum_{k \geq 1} \sqrt{T_k - T_{k-1}} \end{aligned}$$

where in the third line we use  $\sum_{\tau=0}^t \alpha_\tau^2 b_\tau^2 > 4^{k-1}$  which by definition holds for any  $T_{k-1} < t \leq T_k$ .

Thus, we are left to show that  $\sum_{k \geq 1} \sqrt{T_k - T_{k-1}} \leq 2\sqrt{\log T} \sqrt{T}$ . To do so, first notice that the maximal value of  $k$  is bounded as follows,

$$\begin{aligned} 4^{k_{\max}-1} &\leq \sum_{t=0}^{T-1} \alpha_t^2 \\ &\leq \sum_{t=0}^{T-1} (t+1)^2 \\ &\leq T^3 \end{aligned}$$

Thus, assuming  $T \geq 2$  we have  $k_{\max} \leq 3 \log_2 T$ , and therefore,

$$\begin{aligned} \sum_{k \geq 1} \sqrt{T_k - T_{k-1}} &= \sum_{k=1}^{k_{\max}} \sqrt{T_k - T_{k-1}} \\ &\leq \sqrt{k_{\max}} \left( \sum_{k=1}^{k_{\max}} (T_k - T_{k-1}) \right)^{1/2} \\ &\leq \sqrt{3 \log T} (T - T_0)^{1/2} \\ &\leq \sqrt{3 \log T} \sqrt{T}. \end{aligned}$$

where we used  $\|u\|_1 \leq \sqrt{n} \|u\|_2$ ,  $\forall u \in \mathbb{R}^n$  and also  $T_{k_{\max}} = T - 1$ . This established the lemma.  $\square$

## C Proof of Theorem 4.1

*Proof.* For brevity we will not rehearse all of the details which are similar to the proof of the offline setting, but rather only emphasize the differences compared to the analysis of Theorem 3.2. First note the following which is analogous to Lemma B.1,

**Lemma C.1.** *Assume that  $f$  is convex and  $G$ -Lipschitz. Assume that we invoke Algorithm 2 but provide it with noisy gradient estimates (see Eq. (9)) rather than the exact ones. Then for any sequence of non-negative weights  $\{\alpha_t\}_{t \geq 0}$ , and learning rates  $\{\eta_t\}_{t \geq 0}$ , the following holds,*

$$\begin{aligned} & \alpha_t(f(y_{t+1}) - f(z)) \\ & \leq \eta_t \alpha_t^2 \|\tilde{g}_t\|^2 + \eta_t \alpha_t^2 \|\tilde{g}_t\| G + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2) + (\alpha_t^2 - \alpha_t) (f(y_t) - f(y_{t+1})) \\ & \quad + \alpha_t (g_t - \tilde{g}_t) \cdot (z_t - z) \end{aligned}$$

We prove this lemma in Appendix C.1.

Now, focusing on the term  $\alpha_t (g_t - \tilde{g}_t) \cdot (z_t - z)$ , the unbiasedness of  $\tilde{g}_t$  immediately implies,

$$\mathbf{E}[\alpha_t (g_t - \tilde{g}_t) \cdot (z_t - z)] = 0.$$

Ignoring this term and comparing the bound in the above lemma to Lemma B.1, one can see that the expression are identical up to replacing,  $g_t \leftrightarrow \tilde{g}_t$ . This identity in the expressions applies also to the learning rate,  $\eta_t$  (again up to replacing,  $g_t \leftrightarrow \tilde{g}_t$ ). Thus, the exact same analysis as of Lemma B.1 shows that w.p. 1 we have,

$$\sum_{t=0}^{T-1} \alpha_t (f(y_{t+1}) - f(z)) - \sum_{t=0}^{T-1} \alpha_t (g_t - \tilde{g}_t) \cdot (z_t - z) \leq O(GD \sqrt{\log T} \cdot T^{3/2}).$$

Taking expectations and using the above in conjunction with the definition of  $\bar{y}_T$  and Jensen's inequality concludes the proof.  $\square$

### C.1 Proof of Lemma C.1

*Proof.* The proof follows similar lines to the proof of Lemmas B.1 and 3.1. Here we will highlight the changes due to the stochastic setting.

Our starting point is bounding  $\alpha_t (f(x_{t+1}) - f(z))$  which can be decomposed as follows,

$$\begin{aligned} \alpha_t (f(x_{t+1}) - f(z)) & \leq \alpha_t g_t \cdot (x_{t+1} - z) \\ & = \alpha_t \tilde{g}_t \cdot (z_t - z) + \alpha_t g_t \cdot (x_{t+1} - z_t) + \alpha_t (g_t - \tilde{g}_t) \cdot (z_t - z) \end{aligned} \quad (37)$$

Due to the unbiasedness of  $\tilde{g}_t$  then the expectation of the last term  $\alpha_t (g_t - \tilde{g}_t) \cdot (z_t - z)$  is zero. Let us now bound the remaining two terms in the above equation.

**(a) Bounding  $\alpha_t \tilde{g}_t \cdot (z_t - z)$ :** Similarly to the proof of Lemma 3.1 we can show the following to hold (see Eq. (20) in Lemma 3.1),

$$\alpha_t \tilde{g}_t \cdot (z_t - z) \leq \alpha_t^2 \tilde{g}_t \cdot (x_{t+1} - y_{t+1}) - \frac{\alpha_t^2}{2\eta_t} \|x_{t+1} - y_{t+1}\|^2 + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2)$$

Combining the above with  $\|x_{t+1} - y_{t+1}\| = \eta_t \|\tilde{g}_t\|$  implies,

$$\alpha_t \tilde{g}_t \cdot (z_t - z) \leq \eta_t \alpha_t^2 \|\tilde{g}_t\|^2 + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2) \quad (38)$$

**(b) Bounding  $\alpha_t g_t \cdot (x_{t+1} - z_t)$ :** Similarly to the proof of Lemma B.1 we can show the following to hold (see Eq. (34) therein),

$$g_t \cdot (x_{t+1} - z_t) \leq \alpha_t G \eta_t \|\tilde{g}_t\| - (f(y_{t+1}) - f(x_{t+1})) + (\alpha_t - 1) (f(y_t) - f(y_{t+1})) \quad (39)$$



**(c) Bounding  $\alpha_t \cdot (f(y_{t+1}) - f(z))$ :** Combining Equations (38), (39) and (37) we get,

$$\begin{aligned} & \alpha_t (f(x_{t+1}) - f(z)) \\ & \leq \left\{ \eta_t \alpha_t^2 \|\tilde{g}_t\|^2 + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2) \right\} + \alpha_t (g_t - \tilde{g}_t) \cdot (z_t - z) \\ & \quad + (\alpha_t^2 - \alpha_t) (f(y_t) - f(y_{t+1})) + \eta_t \alpha_t^2 \|\tilde{g}_t\| G - \alpha_t (f(y_{t+1}) - f(x_{t+1})) \end{aligned}$$

Re-arranging the above equation and we get,

$$\begin{aligned} & \alpha_t (f(y_{t+1}) - f(z)) \\ & \leq \eta_t \alpha_t^2 \|\tilde{g}_t\|^2 + \eta_t \alpha_t^2 \|\tilde{g}_t\| G + \frac{1}{2\eta_t} (\|z_t - z\|^2 - \|z_{t+1} - z\|^2) + (\alpha_t^2 - \alpha_t) (f(y_t) - f(y_{t+1})) \\ & \quad + \alpha_t (g_t - \tilde{g}_t) \cdot (z_t - z) \end{aligned}$$

which concludes the proof. □

## D Proof of Theorem 4.2

*Proof of Theorem 4.2.* Lets us denote by  $\tilde{g}_t$  the noisy gradients received by AdaGrad upon querying  $x_t$ . In this case, by applying the regret guarantees of AdaGrad, [13], in conjunction to standard online to batch conversion technique, [8], implies,

$$\sum_{t=1}^T \mathbf{E} \left( f(x_t) - \min_{x \in \mathcal{K}} f(x) \right) \leq \mathbf{E} \sqrt{2D^2 \sum_{t=1}^T \|\tilde{g}_t\|^2} \quad (40)$$

Now decomposing,  $\|\tilde{g}_t\| \leq \|g_t\| + \|\tilde{g}_t - g_t\|$ , gives,

$$\sqrt{\sum_{t=1}^T \|\tilde{g}_t\|^2} \leq \sqrt{2 \sum_{t=1}^T \|g_t\|^2 + 2 \sum_{t=1}^T \|\tilde{g}_t - g_t\|^2} \leq \sqrt{2 \sum_{t=1}^T \|g_t\|^2} + \sqrt{2 \sum_{t=1}^T \|\tilde{g}_t - g_t\|^2}.$$

where the first inequality uses  $(a + b)^2 \leq 2a^2 + 2b^2$ , and the second inequality uses  $(a + b)^{1/2} \leq a^{1/2} + b^{1/2}$  for non-negative  $a, b \in \mathbb{R}$ . Combining the above with Eq. (40) and applying Jensen's inequality with respect to the function  $H(u) = \sqrt{u}$ , gives,

$$\begin{aligned} \sum_{t=1}^T \mathbf{E} \left( f(x_t) - \min_{x \in \mathcal{K}} f(x) \right) &\leq 2 \sqrt{D^2 \sum_{t=1}^T \mathbf{E} \|g_t\|^2} + 2 \sqrt{D^2 \sum_{t=1}^T \mathbf{E} \|\tilde{g}_t - g_t\|^2} \\ &\leq 2 \sqrt{2\beta D^2 \sum_{t=1}^T \mathbf{E} \left( f(x_t) - \min_{x \in \mathcal{K}} f(x) \right)} + 2\sqrt{\sigma^2 D^2 T} \end{aligned} \quad (41)$$

the last line uses the lemma below, which holds since we assume  $\mathcal{K}$  contains a global minimum.

**Lemma D.1.** *Let  $F : \mathbb{R}^d \mapsto \mathbb{R}$  be a  $\beta$ -smooth function, and let  $x^* = \arg \min_{x \in \mathbb{R}^d} F(x)$ , then,*

$$\|\nabla F(x)\|^2 \leq 2\beta (F(x) - F(x^*)), \quad \forall x \in \mathbb{R}^d.$$

Eq. (40) enables to show,  $\sum_{t=1}^T \mathbf{E} (f(x_t) - \min_{x \in \mathcal{K}} f(x)) \leq 4\beta D^2 + 2\sigma D\sqrt{T}$ . Combining this together with the definition of  $\bar{x}_T$  and Jensen's inequality concludes the proof.  $\square$

### D.1 Proof of Lemma D.1

*Proof.* The  $\beta$  smoothness of  $F$  means the following to hold  $\forall x, u \in \mathbb{R}^d$ ,

$$F(x + u) \leq F(x) + \nabla F(x)^\top u + \frac{\beta}{2} \|u\|^2.$$

Taking  $u = -\frac{1}{\beta} \nabla F(x)$  we get,

$$F(x + u) \leq F(x) - \frac{1}{\beta} \|\nabla F(x)\|^2 + \frac{1}{2\beta} \|\nabla F(x)\|^2.$$

Thus:

$$\begin{aligned} \|\nabla F(x)\| &\leq \sqrt{2\beta (F(x) - F(x + u))} \\ &\leq \sqrt{2\beta (F(x) - F(x^*))}, \end{aligned}$$

where in the last inequality we used  $F(x^*) \leq F(x + u)$  which holds since  $x^*$  is the *global* minimum.  $\square$

## E Additional Numerical Experiments

Here, we present numerical experiments on the stochastic setting, and on a practical variant that neglects the projection steps.

### E.1 Numerical Experiments on the Stochastic Setting

We consider the same problem setup as in Section 5. Rather than using the exact gradients, we compute the unbiased estimates evaluated by a single data point (i.e. minibatch of size 1) The results are shown in Figure 2.

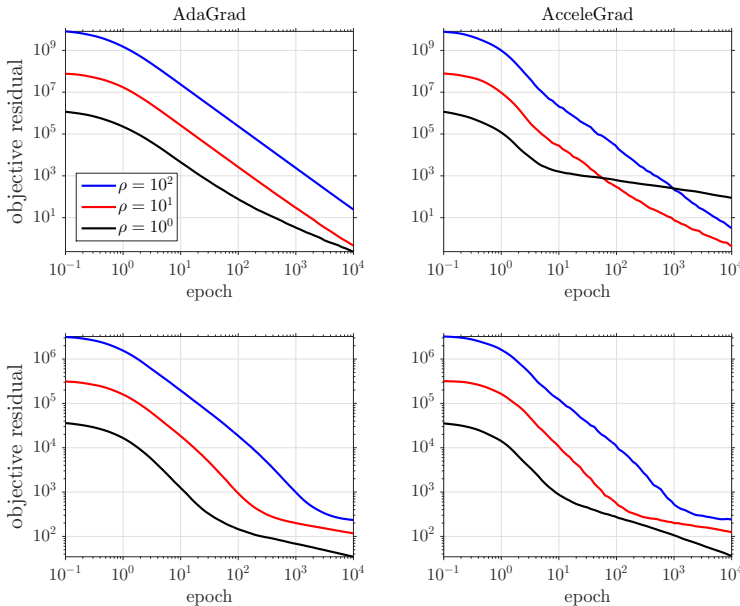


Figure 2: Comparison of AdaGrad and AcceleGrad in stochastic setting for smooth (*top*) and non-smooth (*bottom*) problems. Epoch denotes one full data pass, hence 500 iterations.

AdaGrad and AcceleGrad perform similar empirically for most of the parameter choices. AdaGrad overperforms AcceleGrad only for the smooth problem with  $\rho = 1$ . This behavior is caused by the projection step, and slightly increasing  $D$  cures the problem for AcceleGrad.

Universal gradient methods [27] are based on a line-search technique that relies on the exact first order oracle information. Thus, it is not so surprising that in practice these methods fail upon receiving stochastic feedback, and we therefore do not present their performance.

### E.2 Numerical Experiments Neglecting the Projections

We observed that the methods work well in practice even if we ignore the projection step in the unconstrained setting. In some cases, this simple tweak may even improve the performance. We used the same test setup as in Section 5, and the results are shown in Figures 3 and 4 for the deterministic and stochastic settings respectively. Note that the method works also when we underestimate  $D$ .

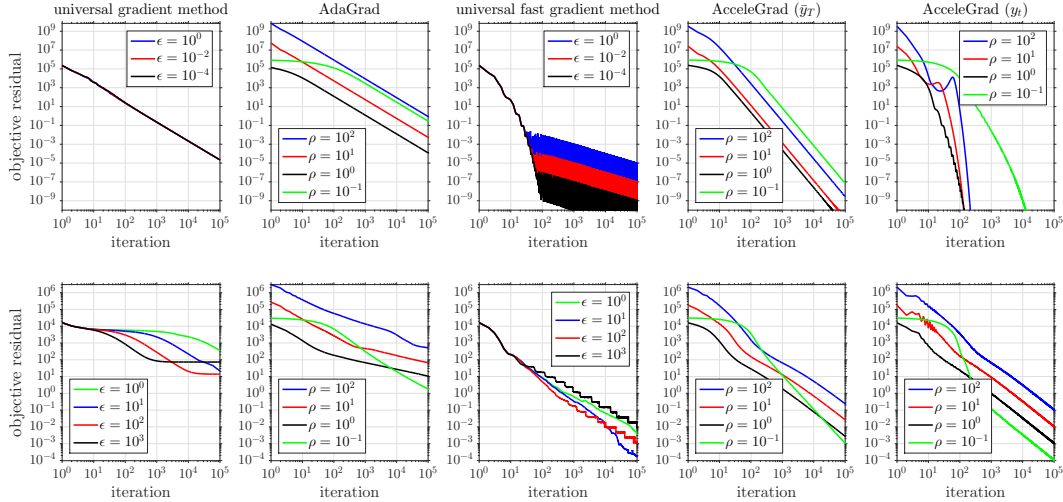


Figure 3: Comparison of universal methods at a smooth (*top*) and a non-smooth (*bottom*) problem. Adaptive methods are tweaked to ignore the projection.

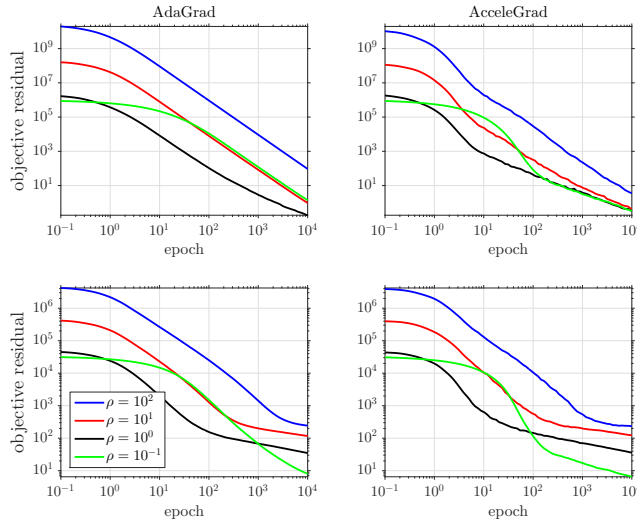


Figure 4: Comparison of AdaGrad and AcceleGrad in stochastic setting for smooth (*top*) and non-smooth (*bottom*) problems. Methods are tweaked to ignore the projection. Epoch denotes one full data pass, hence 500 iterations.

### E.3 Experiments with Large Minibatch

In this section we apply AcceleGrad to a real world stochastic optimization problem and compare its performance with AdaGrad. We examine the effect of minibatch size versus performance. The large minibatch regime is important when one likes to apply SGD using several machines in parallel. This is done by dividing the minibatch computation between the machines. Unfortunately, it is well known that the performance of SGD degrades with the increase of minibatch size  $b$ . Here, we show that AcceleGrad might be more appropriate in this case.

Concretely we consider the RCV1<sup>4</sup> dataset which is a binary labeled set with 20424 datapoints samples and 47366 features. We train a classifier for this dataset using logistic loss (smooth case) as well as using the hinge loss (SVM). We compare the performance of AcceleGrad with AdaGrad.

<sup>4</sup>available in the UCI repository website (<https://archive.ics.uci.edu/ml/>)

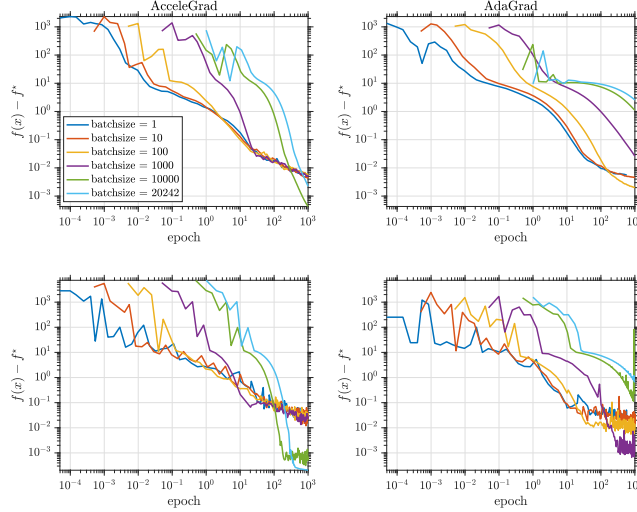


Figure 5: Comparison of AdaGrad and AcceleGrad for logistic regression task using different minibatch sizes. We display the averaged iterates,  $\bar{y}_T$  (*top*), as well as the non-averaged iterates,  $y_t$  (*bottom*). Both methods use the same parameter  $D = 10^4$ .

For each method we examine several minibatch sizes, and observe the performance of each method verses the number of epochs (total number of gradients that we have computed).

The results for logistic regression appear in Figure 6. For AdaGrad we see that the performance degrades as we increase the minibatch size beyond  $b = 1000$ . This actually agrees with theory that predicts a degradation with the increase of  $b$ .

For AcceleGrad we observe an interesting phenomenon: if we aim for a very small error (in this case smaller than  $10^{-2}$ ) then as we increase the minibatch size the performance actually improves. The intuition behind this is the following: upon using small  $b$  the gradients are noisy and both AcceleGrad and AdaGrad will obtain the slow  $\mathcal{O}(1/\sqrt{T})$  rate, where  $T$  is the number of iterations. However, as  $b$  increases the gradients are becoming more accurate and AcceleGrad will obtain a rate approaching  $\mathcal{O}(1/T^2)$  while AdaGrad will approach  $\mathcal{O}(1/T)$  rate. Now note that the number of gradient calculations  $S$ , depends on  $b$  and  $T$  as follows,  $T = S/b$ .

Thus, for small minibatch, both methods will ensure a rate of  $\mathcal{O}(\sqrt{b}/\sqrt{S})$ , which clearly degrades with  $b$ . As  $b$  increases AcceleGrad will obtain a rate approaching  $\mathcal{O}(b^2/S^2)$  while AdaGrad will approach  $\mathcal{O}(b/S)$  rate.

We have observed similar behaviour when train an SVM (i.e., using hinge loss). This can be seen in Figure 5.

Note that we have performed several other experiments with different  $D$  parameters, and also different  $\ell_3$  regularization parameters. In all experiments we have seen the same qualitative behaviour that we describe above.

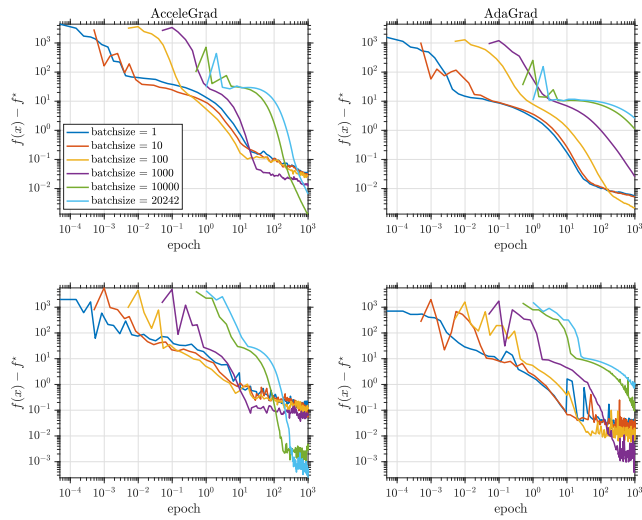


Figure 6: Comparison of AdaGrad and AcceleGrad for training SVM using different minibatch sizes. We display the averaged iterates,  $\bar{y}_T$  (*top*), as well as the non-averaged iterates,  $y_t$  (*bottom*). Both methods use the same parameter  $D = 10^4$ .