GEOADALER: GEOMETRIC INSIGHTS INTO ADAPTIVE STOCHASTIC GRADIENT DESCENT ALGORITHMS

CHINEDU ELEH[†,§], MASUZYO MWANZA[†], EKENE AGUEGBOH[‡], HANS-WERNER VAN WYK[†]

Abstract. The Adam optimization method has achieved remarkable success in addressing contemporary challenges in stochastic optimization. This method falls within the realm of adaptive sub-gradient techniques, yet the underlying geometric principles guiding its performance have remained shrouded in mystery, and have long confounded researchers. In this paper, we introduce GeoAdaLer (Geometric Adaptive Learner), a novel adaptive learning method for stochastic gradient descent optimization, which draws from the geometric properties of the optimization landscape. Beyond emerging as a formidable contender, the proposed method extends the concept of adaptive learning by introducing a geometrically inclined approach that enhances the interpretability and effectiveness in complex optimization scenarios.

Key words. adaptive learning rate, stochastic optimization, convex optimization, large-scale optimization, machine learning

MSC codes. 90C25, 90C15, 90C30, 68Q32, 68T07

1. Introduction.

Stochastic gradient descent (SGD) optimization methods [25, 29] play an important role in various scientific fields. When applied to machine learning algorithms, the objective is to adjust a set of parameters with the goal of optimizing an objective function. This optimization usually involves a series of iterative adjustments made to the parameters in each step as the algorithm progresses [14]. In the vanilla gradient descent approach, the magnitude of the gradient is the predominant annealing factor causing the algorithm to take larger steps when you are away from the optimum and smaller steps when closer to an optimum [27]. This method, however, becomes less effective near an optimal point, necessitating the selection of a smaller learning rate, which in turn affects the speed of convergence. The raw magnitude of the gradient does not always align with the optimal descent step size, thus necessitating a manually chosen learning rate. If the learning rate is too big, overshooting may occur and convergence rate is slow if the learning rate is too small. This challenge has led to the development of adaptive learning algorithms [23]. In this paper, we explore gradient descent algorithms that optimize the objective function with emphasis on the update rule. In this context, we focus on the update rule by breaking it into three components: the learning rate, the annealing factor, and the descent direction. This approach allows us to evaluate the effectiveness of an algorithm by examining the impact of its learning rate, annealing factor, and descent direction on the optimization process.

In this paper, we propose GeoAdaLer (short for Geometric Adaptive Learner), a new adaptive learning method for SGD optimization that is based on the geometric properties of the objective landscape. We use cosine of θ (where θ is the acute angle between the normal to the tangent hyperplane and the horizontal hyperplane) as an annealing factor, which takes values close to zero when the optimization is traversing points close to an optimum and close to one for points far away from an optimum.

[†]Department of Mathematics and Statistics, Auburn University, Auburn, AL 36849, USA (cae0027@auburn.edu, mzm0183@auburn.edu, hzv0008@auburn.edu).

[‡]Department of Applied Economics, Auburn University, Auburn, AL 36849, USA (esa0013@auburn.edu).

[§]Corresponding author (cae0027@auburn.edu).

Our method has surprising similarities to other adaptive learning methods.

Some of the advantages of GeoAdaLer is that it introduces a geometric approach for the annealing factor and outperforms standard SGD optimization methods due to the cosine of θ. Through both theoretical analysis and empirical observation, we identified similarities between our proposed method and other adaptive learning techniques. Our method enhances the understanding of existing algorithms and opens up more opportunities for geometric interpretability of how the algorithms traverse the objective manifold.

Additionally, we analyze the convergence of GeoAdaLer. We frame the optimization process as a fixed-point problem and split the analysis into deterministic and stochastic cases. Under the deterministic framework, we assume convexity and the existence of a finite optimal value. We utilize the Lipschitz continuity property of the gradient to establish upper bounds of convergence. We further employ the cocoercivity and quadratic upper bound properties to establish the lower bounds of convergence [34]. Under the stochastic framework, we employ the regret function, which measures the overall difference between our method and the known optimum point, ensuring that as time tends to infinity, the regret function over time tends to zero. By ensuring that the upper bound of the regret function goes to zero, we determine the overall convergence of the stochastic method to an optimum point. Both the deterministic and stochastic analyses demonstrate the robustness of GeoAdaLer and enhance our understanding of its practical applications.

2. Related Work. In the field of adaptive stochastic gradient descent algorithms, we continue to see improvements, often due to the need to address shortcomings of previous methods. The pioneering approach is AdaGrad, which focuses on the concept of per-parameter adaptive learning rates. The foundation of AdaGrad is also the source of its limitation: the monotonic accumulation of squared gradients that could prematurely stifle learning rates [37, 8].

Subsequent optimizers like AdaDelta, RMSprop, and the popular Adam sought to address this issue [14, 33, 37]. AdaDelta introduced a decaying average of past squared gradients, while RMSprop utilized a similar exponential decay mechanism to limit the aggressive reduction in learning rates. Adam combined RMSprop's adaptive learning rates with the concept of momentum for smoother updates. However, Adam has been challenged by AMSGrad which modifies the algorithm in order to imbue it with "long term memory" which addresses issues with sub-optimal convergence under specific conditions and improves empirical performance [24]. As pointed out by [21, 31, 38], the assumptions on the hyperparameters of Adam were made before constructing the counter examples in [24]. Insights gained from these examples are invaluable for the ongoing exploration of stochastic gradient descent algorithms.

The literature on adaptive optimization algorithms for SGD reveals that each algorithm addresses the problems evident in its predecessors. The iterative approach, while successful in many ways, arguably emphasizes the lack of fundamental intuition regarding the dynamics of the AdaGrad family of optimizers.

GeoAdaLer is a novel adaptive learning method for SGD, employing the properties of the objective landscape. The innovative idea behind the GeoAdaLer approach is that the acute angle between the normal to the tangent plane at x and the horizontal plane conveys significant curvature-related information. This information could potentially recover the power of second order methods, which are not feasible in large-scale machine learning optimization. An understanding of this geometric approach to analyze adaptive methods for SGD can shed new light on the behavior of other algo-

rithms in the AdaGrad family, potentially revealing the rationale for their strengths and weaknesses.

By discussing the geometric implications of adaptive step processes, we are able to potentially come up with optimizers that:

- **have more optimal annealing**: A geometric perspective that informs strategies to control learning rate decay more effectively, preventing premature convergence or extensively slow convergence.
- **provide parameter-sensitive adaptivity**: The geometry reveals how to tailor updates for individual parameters in a more principled manner.
- **increase robustness**: Understanding the geometric implications leads to optimizers that are less dependent on sensitive hyper-parameter tuning.

In essence, a geometric framework promises to move beyond the reactive development pattern, allowing us to proactively design adaptive optimizers that address the core issues in the AdaGrad family with greater intuition and foresight.

3. Mathematical Formulation.

3.1. Deterministic Optimization.

Consider the minimization of the convex objective functional $f : \mathbb{R}^n \to \mathbb{R}$ using the gradient descent algorithm. The vanilla gradient descent (GD) algorithm that maximizes the benefit of *gradient annealing* for smooth functions is

$$x_{t+1} = x_t + \delta x_t \tag{3.1}$$

where $\delta x_t = -\gamma g_t$ and g_t is the gradient of the objective function at time t. This is the update step and is largely responsible for how far a step is taken in the descent direction. To see its real contribution, we decompose it further into

$$\delta x_t = -\gamma \|g_t\| \overline{g_t} \tag{3.2}$$

where $\gamma > 0$ is the learning rate which is responsible for manually scaling the step size, $\|g_t\|$ is the annealing factor, and $-\overline{g_t}$ is the unit vector in the descent direction.

It is established that the annealing factor tends to be sub-optimal and can cause overshooting which we need to compensate for by applying a smaller learning rate, increasing convergence time [23]. To combat this issue, we propose an annealing factor based on the cosine of θ_t, where θ_t is the acute angle between the normal to the tangent hyperplane and the horizontal hyperplane at iteration t. As shown in Figure 1, θ_t holds a vital information about the location of the current gradient step on the objective function. We harness this information using the cosine of θ_t and prove the following theorem.

THEOREM 3.1 (Geohess). *Let θ be the acute angle between the normal to an objective function $f : \mathbb{R}^n \to \mathbb{R}$ which is differentiable at x. Let $\|\cdot\|$ be the norm induced by the inner product on $\mathbb{R}^n$. Then*

$$\cos\theta = \frac{\|\nabla f(x)\|}{\sqrt{\|\nabla f(x)\|^2 + 1}}. \tag{3.3}$$

We call Theorem 3.1 *Geohess* since this formulation mimics the curvature information traditionally found in the full hessian matrix.

Proof. Let x_i be an arbitrary point in $\mathbb{R}^n$. Then $[-\nabla f(x_i), 1]^T$ is orthogonal to $[x - x_i, y - f(x_i)]$ where (x, y) lies on the tangent hyperplane to f at x_i. i.e.,

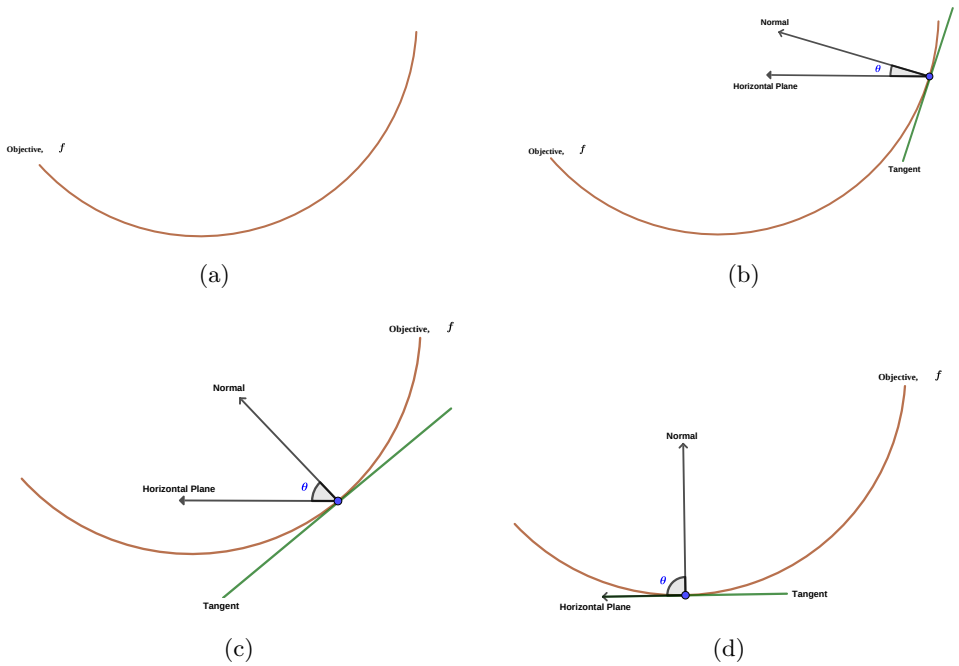


Fig. 1: Geometric view of θ as the GeoAdaLer step traverses the objective function.

$[-\nabla f(x_i), 1]^T$ is normal to $[x-x_i, y-f(x_i)]$. Let θ be the angle this normal makes with the horizontal hyperplane at the point $(x_i, f(x_i))$ in the descent direction $-\nabla f(x_i)$, i.e., a vector parallel to $[-\nabla f(x_i), 0]$. Then by vector calculus,

$$\cos\theta = \frac{[-\nabla f(x_i), 1]^T \cdot [-\nabla f(x_i), 0]^T}{\|[-\nabla f(x_i), 1]^T\| \|[-\nabla f(x_i), 0]^T\|} = \frac{\|\nabla f(x_i)\|}{\sqrt{\|\nabla f(x_i)\|^2 + 1}} \tag{3.4}$$

as required. ∎

The proposed GeoAdaLer update step is as follows:

$$\delta x_t = -\gamma\, \overline{g_t} \cos\theta_t = -\gamma \frac{\|g_t\|}{\sqrt{\|g_t\|^2 + 1}} \overline{g_t} Provide \tag{3.5}$$

Equation (3.5) follows from the Geohess theorem (Theorem 3.1). In equation (3.5), if g_t is zero, then the update step is zero and algorithm has converged. If g_t is not zero, then equation (3.5) reduces to

$$\delta x_t = -\gamma \frac{g_t}{\sqrt{\|g_t\|^2 + 1}}$$

3.2. Properties of GeoAdaLer Annealing Factor. The acute angle θ_t depends on g_t, and therefore, the annealing factor possesses the following properties as $\|g_t\| \to 0$ and as $\|g_t\| \to \infty$ respectively:

1. $\dfrac{\|g_t\|}{\sqrt{\|g_t\|^2+1}} \to 1$ 2. $\dfrac{\|g_t\|}{\sqrt{\|g_t\|^2+1}} \to 0$

In practice, we do not want $\|g_t\| \to \infty$ but as long as the magnitude of the gradient is large, the sufficiency in item 1 is guaranteed. Of utmost importance is item 2.

The GeoAdaLer algorithm is presented in Algorithm 3.1. In the deterministic setting, we set $\beta = 0$.

3.3. Stochastic Optimization. Online learning and stochastic optimization are closely linked and can be essentially used interchangeably [8, 5]. In online learning, the learner iteratively predicts a point $x_t \in X \subseteq \mathbb{R}^n$, often representing a weight vector that assigns importance values to different features. The objective of the learner is to

$$\underset{x}{\text{minimize}}\, \mathop{\mathbb{E}}_{\xi}[f(x,\xi)],$$

which is too difficult to predict [10]. Instead, we minimize regret, defined as the difference between the learner's cumulative loss and the cumulative loss of the best fixed predictor in hindsight, $x^* \in X$, chosen from a closed convex set (details Section 5.2). This is performed across a sequence of convex loss functions $\{f_1, f_2, \cdots\}$, where each function represents feedback or loss information available at step t [39, 30, 10].

Geometrically, gradients are vectors and have directions. Under a suitable distribution, a convex hull of these gradients approximates the true gradient for stochastic optimizations. A common and intuitive approach to approximating the expected gradient is by employing an exponential moving average (EMA). As is the practice in literature [12, 14], we use a momentum-like term to replace the instantaneous gradient, thereby mitigating stochastic fluctuations and revealing underlying trends in the gradient values. That is,

$$m_{t+1} = \beta m_t + (1-\beta) g_t. \tag{3.6}$$

where $\beta \in [0, 1)$. We have used $g_t = \nabla f_t(x_t)$ for notational convenience. The term β is constructed as a weighted average of the historical gradients and the current gradient. This modification enhances our ability to approximate the true underlying gradient more effectively in time. It also has the ability to update and learn as new streams of data are observed.

In Algorithm 3.1, we present a pseudo-code of the proposed GeoAdaLer learning method.

3.3.1. GeoAdaMax. Due to stochasticity and varying frequencies of occurrence of certain model inputs, such as in deep neural networks, adaptive stochastic gradient descent methods sometimes encounter issues with non-increasing squared gradients. For instance, consider the convex objective function presented in [24] over the domain $[-1, 1]$, defined as follows:

$$f_t(x) = \begin{cases} Cx, & \text{if } t \mod 3 = 1 \\ -x, & \text{otherwise,} \end{cases}$$

where $C > 2$. In such scenarios, the adaptive step employed by the optimizer can still function effectively as a form of annealing, but the vanilla adaptive step leads to a suboptimality.

To address this problem, an approach was developed by [24], which involves retaining the maximum of the normalizing denominator over iterations. This approach

Algorithm 3.1 GeoAdaLer

Require: γ : Learning rate
Require: β : Exponential decay rate for weighted average
Require: f_t : Stochastic objective function
Require: x_0 : Initial parameter vector
  $t \leftarrow 0$
  while x_t not converged **do**
    $g_t \leftarrow \nabla f_t(x_t)$ (Get gradients w.r.t stochastic objective)
    if $t = 0$ **then**
      $m_t \leftarrow g_t$ (initial weighted average)
    else
      $m_t = \beta m_{t-1} + (1-\beta) g_t$ (Updated weighted average)
    end if
    $x_{t+1} = x_t - \gamma \cdot m_t / (\sqrt{\|m_t\|^2 + 1})$ (Update parameters)
    $t \leftarrow t + 1$
  end while
  return x_t

reduces to a self scaling SGD with momentum [24]. By implementing a similar idea, we observe related results for GeoAdaLer. We call this method GeoAdaMax, indicating the use of maximum of the variance term.

GeoAdaMax dynamically adjusts the denominator using the largest historical value of the EMA. When this denominator remains unchanged, the update scale reflects its historical maximum, preserving proportionality in all future updates. This mechanism fine-tunes step sizes while maintaining alignment with past gradient magnitudes. The summary of the algorithm is presented in Algorithm 3.2

Geometrically, this is akin to increasing the angle between the normal and the horizontal plane by using a vector different from the normal. This leads to relatively smaller step sizes than if the original angle were used. Theorem 3.2 shows that indeed, the acute angle θ is increased.

THEOREM 3.2. *Let θ_t be the acute angle between the normal and the horizontal hyperplanes at the current iteration and let $\hat{\theta}_t$ be the acute angle between the horizontal hyperplane and the normal that maximizes the norm up to the current iteration, t. Then*

$$\theta_t \leq \hat{\theta}_t.$$

Proof. By Algorithm 3.1 and Theorem 3.1,

$$\cos\theta_t = \frac{\|m_t\|}{\sqrt{\|m_t\|^2 + 1}} \geq \frac{\|m_t\|}{\sqrt{\max_t \|m_t\|^2 + 1}} = \cos\hat{\theta}_t$$

Applying the inverse cosine function on both sides gives the desired result since $\cos^{-1}$ is monotone decreasing on $[0, 1]$. ∎

4. Relationship to the AdaGrad Family. The AdaGrad family of methods adjusts learning rates based on the accumulation of past gradients. These methods dynamically adapt the step size during optimization to handle varying gradient magnitudes across dimensions. For a given time step t, the learning rate adjustment in

Algorithm 3.2 GeoAdaMax

Require: γ: Learning rate
Require: β: Exponential decay rate for weighted average
Require: f_t: Stochastic objective function
Require: x_0: Initial parameter vector
  $t \leftarrow 0$
  $u_t \leftarrow 0$
  while x_t not converged **do**
    $g_t \leftarrow \nabla f_t(x_t)$ (Get gradients w.r.t stochastic objective)
    if $t = 0$ **then**
      $m_t \leftarrow g_t$ (initial weighted average)
    else
      $m_t = \beta m_{t-1} + (1-\beta) g_t$ (Update weighted average)
    end if
    $u_t = \max(\|m_t\|^2 + 1, u_{t-1})$
    $x_{t+1} = x_t - \gamma \cdot m_t/(\sqrt{u_t})$ (Update parameters)
    $t \leftarrow t + 1$
  end while
  return x_t

these methods depends on the accumulated gradient information, which we represent as G_t. Below is a summary of how G_t is defined for the main methods in the AdaGrad family:

AdaGrad: For AdaGrad, G_t is given as

$$G_t = \sum_{i=1}^{t} g_i^2$$

where g_i is the gradient at step i. AdaGrad accumulates the squared gradients over time, leading to decreasing learning rates [8].

RMSProp: The G_t step in RMSProp involves exponential moving average and is given as

$$G_t = \beta G_{t-1} + (1-\beta) g_t^2.$$

RMSProp introduces an exponential decay factor β, preventing the learning rate from decreasing too quickly by giving more weight to recent gradients [12].

AdaGrad and RMSProp modify the learning rate as follows:

$$x_{t+1} = x_t - \frac{\gamma}{\sqrt{G_t + \epsilon}} g_t,$$

where η is the global learning rate and ϵ is a small constant for numerical stability.

Adam: In the case of Adam, an adjustment is not only made to the squared gradients but also to the gradients where a bias-corrected moving average is applied to each:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$
$$G_t = \beta_2 G_{t-1} + (1-\beta_2) g_t^2$$

where m_t and G_t are the moving average of the gradients and the squared gradients respective, $\hat{m}_t = m_t/(1-\beta_1^t)$ and $\hat{G}_t = G_t/(1-\beta_2^t)$ are the bias-corrected moving

average of gradients and squared gradients respectively. Adam combines the ideas of momentum and adaptive learning rates for smoother updates [14].

The Adam's update rule therefore is as follows:

$$x_{t+1} = x_t - \frac{\gamma}{\sqrt{\hat{G}_t} + \epsilon}\hat{m}_t,$$

4.1. GeoAdaLer in Comparison. The core of GeoAdaLer's update is based on the cosine of the angle θ between the normal to the gradient and the horizontal hyperplane. It effectively uses the geometry of the optimization landscape to inform its adaptivity. Hence, with an EMA update

$$m_t = \beta m_{t-1} + (1-\beta)g_t,$$

GeoAdaLer updates as:

$$x_{t+1} = x_t - \frac{\gamma}{\sqrt{\|m_t\|^2 + 1}}m_t.$$

As shown above, GeoAdaLer's update is similar to that of the AdaGrad family with some function of the squared gradient playing a big role in the optimization step. The main differences include:

(a) The use of norm-based scaler
(b) The stability term in GeoAdaLer is naturally derived from the choice of the reference plane, for example with the choice of the normal to the gradient, our scaler produces a stability term of 1.
(c) There is only one moving average which is used in the estimation of the gradient and the function of the squared gradients
(d) Directly comparing to Adam, we also agree that G_t is always bigger for Adam since by Jensen's inequality,

$$\left[\beta m_{t-1} + (1-\beta)g_t\right]^2 \leq \beta m_{t-1}^2 + (1-\beta)g_t^2,$$

where the LHS G_t is for GeoAdaLer and the RHS for Adam.

We remark that all of these differences stem from the geometric intuition behind GeoAdaLer, which we believe lends itself to better understanding of the optimization paths as well as interpreting optimal values. Also, as noted in [35], the use of norm-based adaptivity ensures GeoAdaLer is robust to its hyperparameters.

5. Convergence Analysis.

5.1. Deterministic Setting. We analyze the convergence of GeoAdaLer, first for the deterministic case, and then for the stochastic setting. Our setup remains the same, namely:

$$\underset{x}{\text{minimize}}\, f(x) \tag{5.1}$$

using the new adaptive gradient descent (GeoAdaLer) method where $f : \mathbb{R}^n \to \mathbb{R}$ is an objective function. We recast the optimization problem (5.1) into a fixed point iteration. Let $T : \mathbb{R}^n \to \mathbb{R}^n$ be a nonlinear operator defined as:

$$Tx = \left(I - \gamma\frac{\nabla f}{\sqrt{\|\nabla f\|^2 + 1}}\right)(x) \tag{5.2}$$

where I is the identity map.

THEOREM 5.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and ∇f Lipschitz continuous with $\gamma \leq \frac{1}{L}$ where L is the Lipschitz constant for ∇f. Assume f attains an optimal value at $x^* = \arg\min_x f(x)$. Then T defined in (5.2) is a contraction map with contraction parameter $\alpha = \sqrt{1 + \gamma^2 L_G^2 - 2\gamma \frac{L_G^2}{L}} < 1$ where L_G is the Lipschitz constant for $\frac{\nabla f}{\sqrt{\|\nabla f\|^2 + 1}}$ and $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^n$. That is*

$$\|Tx - Ty\| \leq \alpha \|x - y\|. \tag{5.3}$$

A critical component of Theorem 5.1 involves demonstrating that the mapping T is a contraction. This allows us to invoke the Banach Fixed Point Theorem, which asserts that any contraction mapping on a complete metric space possesses a unique fixed point [6, 28, 32, 11]. By iteratively applying the contraction map, starting from an initial point, we ensure convergence to this fixed point. The recursive iterations converge to the unique fixed point with a geometric rate of convergence α. This fixed point corresponds to the minimizer we are seeking [3]. We remark that Theorem 5.1 holds if we replace gradient with subgradient.

5.2. Stochastic Setting. We examine the convergence properties of the GeoAdaLer optimizer within the framework of online learning, as originally introduced in [39]. This framework involves a sequence of convex cost functions $\{f_1, f_2, \ldots, f_T\}$, each of which becomes known only at its respective timestep. The objective at each step t is to estimate the parameter x_t and evaluate it using the newly revealed cost function f_t. Given the unpredictable nature of the sequence, we assess the performance of our algorithm by computing the regret. Regret is defined as the cumulative sum of the differences between the online predictions $f_t(x_t)$ and the optimal fixed parameter $f_t(x^*)$ within a feasible set X across all previous time-steps. Specifically, the regret is defined as follows [39, 14]:

$$R(T) = \sum_{t=1}^{T} \left(f_t(x_t) - f_t(x^*)\right) \tag{5.4}$$

where the optimal parameter x^* is determined by $x^* = \arg\min_{x \in X} \sum_{t=1}^{T} f_t(x)$. We demonstrate that GeoAdaLer achieves a regret bound of $O(\sqrt{T})$, with a detailed proof provided in the appendix C. This result aligns with the best known bounds for the general convex online learning problem. We carry over all notations from the deterministic setting. Assuming the learning rate γ_t is of order $O(t^{-1/2})$ and β_t is exponentially decaying with exponential constant λ very close to 1, we obtain the following regret bounds for online learning with GeoAdaLer algorithm.

THEOREM 5.2. *For all $x \in \mathbb{R}^n$, and $t \leq T$, assume f_t is convex and the gradient norm $\|\nabla f_t(x)\| \leq G$. Let $\gamma_t = \frac{\gamma}{\sqrt{t}}$, $\beta_t = \beta\lambda^t$, $\lambda \in (0, 1)$, and $\beta \in [0, 1)$. For any $k \in \{1, \ldots, T\}$, the separation between any point x_k generated by GeoAdaLer and the minimizer of an offline objective computed after all data is known is bounded as $\|x_k - x^*\| \leq G$. Then, for any $T \geq 1$, GeoAdaLer Algorithm achieves the regret bound:*

$$R(T) \leq \frac{D^2\sqrt{G^2+1}\sqrt{T} + G(2\sqrt{T} - 1)}{2(1-\beta)} + \frac{DG\beta(1-\lambda^T)}{(1-\beta)(1-\lambda)} \tag{5.5}$$

From Theorem 5.2, we observe that the GeoAdaLer algorithm achieves a sublinear regret bound of $O(\sqrt{T})$ over T iterations, which is consistent with the results typically

found in the literature for similar algorithms. Our proof, akin to the approach taken in the Adam algorithm [14], relies significantly on the decay of β_t to ensure convergence.

In contrast to other adaptive stochastic gradient descent methods that adapts the current employs norm based scaling. This approach ensures that each parameter benefits from the collective dynamics of all parameters at the current time while retaining historical information in the exponential moving average of the gradients used for the adaptation. We remark that the convergence analysis for GeoAdaMax follows trivially from Theorem 5.2.

COROLLARY 5.3. *Assume that for any $x \in \mathbb{R}^n$, the function f_t is convex and satisfies the gradient bounds $\|\nabla f_t(x)\|_2 \leq G$. Also, assume that the distance between any parameter x_k generated by GeoAdaLer Algorithm and the minimizer of an offline objective computed after all data is known is bounded, namely $\|x_k - x^*\|_2 \leq D$ for any $k \in \{1, \cdots, T\}$. Then, GeoAdaLer achieves the following guarantee for all $T \geq 1$:*

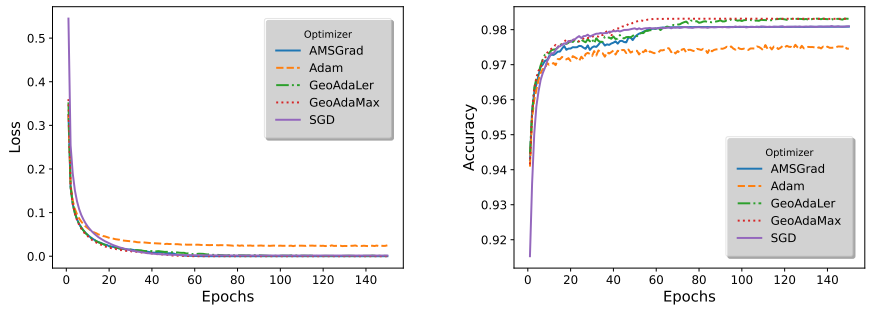
$$\limsup_{T \to \infty} \frac{R(T)}{T} \leq 0$$

The corollary is derived by dividing the result in Theorem 5.2 by T and applying the limit superior operation to both sides of the inequality 5.5. It is important to note that when $R(T)$ yields a negative value, it indicates a favorable performance of the iterates produced by the GeoAdaLer algorithm. Specifically, such results suggest that the algorithm's execution leads to an expected loss that is lower than that of the best possible offline algorithm, which has full foresight of all cost functions and selects for a single optimal vector as proposed in [39].

6. Experiments. We compare GeoAdaLer to other algorithms such as Adam, AMSGrad, and SGD with momentum. This comparison aims to assess how the geometric approach performs against these popular methods on the standard CIFAR-10 and MNIST datasets. Additionally, we examine how key hyperparameters influence the convergence of GeoAdaLer across different datasets.

The hyperparameter settings for the algorithms are detailed as follows. These settings represent default values unless otherwise specified as recommended in the literature or considered standard for their respective packages. For SGD, the learning rate was set to 0.01, momentum to 0.9 and dampening to 0.9. For Adam & AMSGrad, the learning rate was set to 0.001, β_1 to 0.9 and β_2 to 0.99. All CPU calculations were performed on an AMD Ryzen 8-core CPU, and GPU calculations were conducted on a NVIDIA 3080ti.

6.1. MNIST Dataset. In the MNIST [18] experiment we trained a fully connected feed forward neural networks . It consists of three fully connected layers: the first layer takes in 784 input features (flattened 28x28 grayscale images) and outputs 128 features, the second layer reduces these to 64 features, and the final layer outputs 10 logits corresponding to class scores. Each of the first two layers is followed by a ReLU activation function. The final layer provides raw class scores. Using cross-entropy loss, GeoAdaLer and GeoAdaMax algorithms alongside the baseline optimizers were executed for 150 epochs and for 30 different weights initializations. Figure 2 and Table 1 illustrate the averaged results on the test dataset. GeoAdaLer demonstrates comparable initial performance to algorithms such as Adam and SGD, yet it achieves better long-run performance, converging to a higher validation accuracy. GeoAdaMax further improved this performance by faster convergence.



(a) Training loss (b) Validation Accuracy

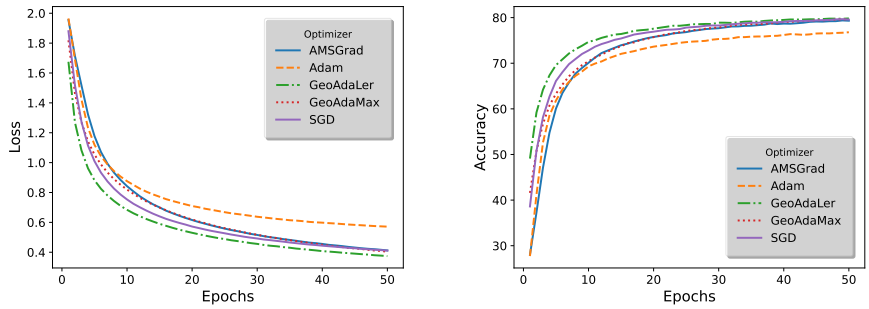
Fig. 2: MNIST

6.2. CIFAR-10 Dataset. In our CIFAR-10 [16] experiments, we run GeoAdaLer, GoeAdaMax and the baseline optimizers for 50 epochs on a network consisting of six convolutional layers with 3x3 kernels and padding, progressively increasing in sizes of $32, 32, 64, 64, 128$ and 128 filters. Each convolution is followed by a batch normalization layer and a ReLU activation. After every two convolutional layers, max-pooling with a 2×2 kernel is applied to reduce spatial dimensions, followed by dropout layers with rates $0.2, 0.3$ and 0.4. The output from the convolutional block, consisting of 128 filters with a 4x4 spatial size, is flattened and passed to two fully connected layers: the first reduces the feature size to 128 with a ReLU activation and a 0.5 dropout, and the second outputs 10 logits corresponding to class scores which are passed to a softmax function. The model is trained using cross-entropy loss and re-run 30 times for each optimizer with different initializations of the weights. The averaged results on test data are shown in Figure 3 and Table 2. GeoAdaLer shows early run performance comparable to baseline optimizers and occasionally exceeds them in validation accuracy. Its long run performance was only matched by Adam and GeoAdaMax. The consistent performance of GeoAdaLer in various runs reinforces the value of incorporating a geometric perspective into its design.

In the CIFAR-10 [16] experiment, we trained a convolutional neural network (CNN) model designed specifically for image data with RGB channels. The architecture consists of six convolutional layers, progressively increasing in feature map depth (32, 64, and 128 channels), followed by max-pooling and dropout layers to reduce overfitting and improve generalization. Each convolutional layer is followed by batch normalization to stabilize learning and expedite convergence [13]. Finally, two fully connected (linear) layers map the feature space to the 10 CIFAR-10 class scores, following common practices for classification in CNN architectures [17].

We train the model using cross-entropy loss for multi-class classification [9], and all adaptive gradient descent algorithms (including GeoAdaLer and GeoAdaMax) were executed for 100 epochs. To account for random initializations, we average the losses and accuracies over 30 different initializations. Model results are illustrated in Figure 3 and Table 2. GeoAdaLer demonstrates initial performance on par with other adaptive optimizers, such as Adam [14] and SGD [19], but achieved superior long-term accuracy, converging to higher validation accuracy. GeoAdaMax provided

further benefits, resulting in faster convergence and smoother training dynamics due to its stability near the optimal point.



(a) Training loss (b) Validation Accuracy

Fig. 3: CIFAR 10 Dataset

Table 1: MNIST Final Accuracy

Optimizer	Accuracy
GeoAdaLer	0.9831
GeoAdaMax	0.9831
Adam	0.9746
AMSGrad	0.9809
SGD	0.9810

Table 2: CIFAR Final Accuracy

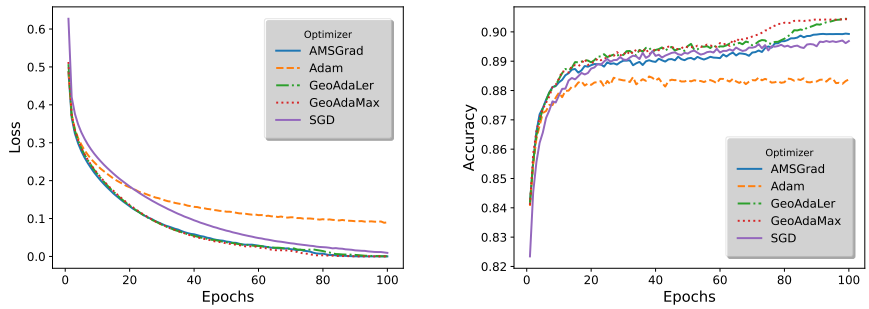
Optimizer	Accuracy
GeoAdaLer	0.7982
GeoAdaMax	0.7962
Adam	0.7679
AMSGrad	0.7932
SGD	0.7957

Table 3: Fashion MNIST Final Accuracy

Optimizer	Accuracy
GeoAdaLer	0.9044
GeoAdaMax	0.9042
Adam	0.8838
AMSGrad	0.8993
SGD	0.8969

6.3. Fashion MNIST. In the Fashion MNIST [36] experiment we train a fully connected feed forward neural networks. It consists of three fully connected layers: the first layer takes in 784 input features (flattened 28×28 grayscale images) and outputs 512 features, the second layer reduces these to 256 features, and the final layer outputs 10 logits corresponding to class scores. Each of the first two layers is followed by a ReLU activation function. The final layer provides raw class scores. Using cross-entropy loss, GeoAdaLer and GeoAdaMax algorithms alongside the baseline

optimizers were executed for 100 epochs and for 30 different weights initializations. Figure 4 and Table 3 illustrate the averaged results on the test dataset. GeoAdaLer once again shows comparable results to that demonstrated by the other benchmark algorithms with GeoAdaMax showing further improvement and faster convergence on the Fashion MNIST dataset.



(a) Training loss (b) Validation Accuracy

Fig. 4: Fashion MNIST Dataset

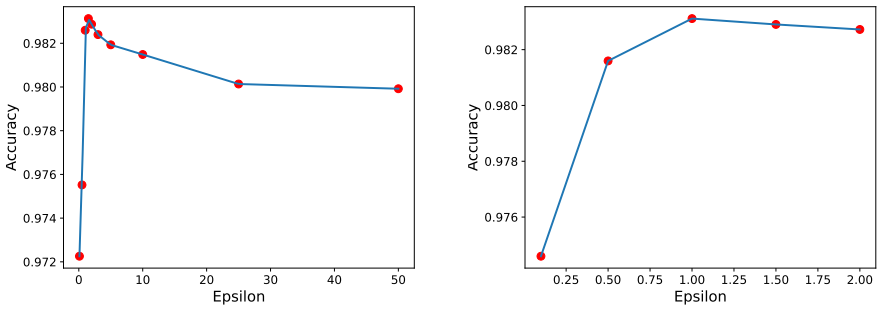
6.4. Alternative Normal Plane Vectors. By introducing a hyper-parameter ϵ in the update rule, we can explore different vectors within the normal plane:

$$x_{t+1} = x_t - \frac{\gamma}{\sqrt{\|m_t\|^2 + \epsilon}} m_t.$$

This approach allows us to select vectors with varying angles relative to the horizontal plane. This investigation stems from observing how GeoAdaMax modifies the effective angles by maximizing the denominator of the update.

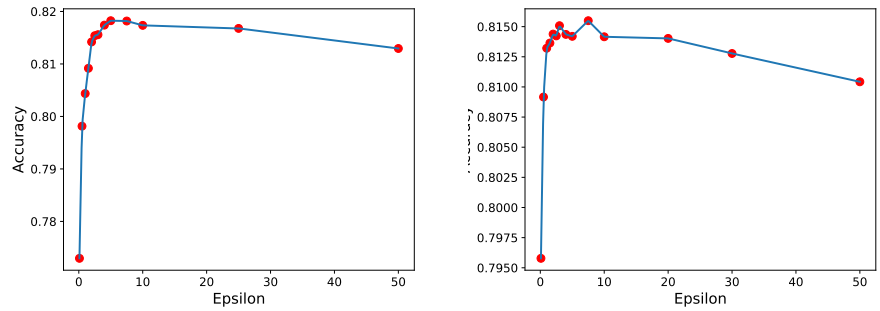
In this experiment, we compared GeoAdaLer and GeoAdaMax on the MNIST (5) and CIFAR-10 (6) datasets using 30 different weight initializations for selected values of ϵ, all other parameters were as mentioned above in their individual experiments. The mean accuracies for each ϵ value are plotted versus the value of ϵ. The results indicate that while the normal vector may not always be the most optimal choice, the optimal ϵ value tends to be close to the default associated with the normal vector. although some improvement do exist through the use of larger values of epsilon it is dependent on the data and not constant through all the experiments.

7. Conclusion. In this paper, we investigate the adaptive stochastic gradient descent algorithm and propose a geometric approach where the normal vector to the tangent hyperplane plays a crucial role in providing curvature-like information. We call this approach GeoAdaLer, and we show that it is derived from a fundamental understanding of optimization geometry. We present theoretical proof for both deterministic and stochastic settings. Empirically, we find that GeoAdaLer is competitively comparable with other optimization techniques. Under certain conditions, it offers better performance and stability. GeoAdaLer provides a general geometric framework applicable to most, if not all, large-scale adaptive gradient-based optimization methods. We believe that this presents a significant step towards the development of interpretable machine learning algorithms through the lens of optimization.



(a) GeoAdaLer Accuracy (b) GeoAdaMax Accuracy

Fig. 5: MNIST Dataset



(a) GeoAdaLer Accuracy (b) GeoAdaMax Accuracy

Fig. 6: CIFAR 10 Dataset

8. Data and Code Availability. Our codebase and the associated datasets used in our experiments are available in an open-source repository on GitHub: https://github.com/Masuzyo/Geoadaler.

REFERENCES

[1] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, Cham, 2 ed., 2017, https://doi.org/10.1007/978-3-319-48311-5.

[2] S. Bock, J. Goppold, and M. Weiss, *An improvement of the convergence proof of the ADAM-optimizer*, 2018, https://doi.org/10.48550/arXiv.1804.10587, https://arxiv.org/abs/1804.10587, https://arxiv.org/abs/1804.10587. arXiv preprint arXiv:1804.10587.

[3] S. Boyd and E. K. Ryu, *A primer on Monotone Operator Methods (survey)*, Applied and Computational Mathematics, 15 (2016), pp. 3–43.

[4] H. Brézis, *Opérateurs maximaux monotones et semi-groupes de contractions dans les espaces de Hilbert*, vol. 5 of North-Holland Mathematics Studies, North-Holland Publishing Company, Amsterdam, 1973. Notas de Matemática, Vol. 50.

[5] N. Cesa-Bianchi, A. Conconi, and C. Gentile, *On the generalization ability of on-line learning algorithms*, IEEE Transactions on Information Theory, 50 (2004), pp. 2050–2057.

[6] C. Chidume, *Geometric Properties of Banach Spaces and Nonlinear Iterations*, Springer, Berlin, 2009.
[7] O. Chidume, *Foundations of mathematical real analysis: Computer science mathematical analysis*, 2019.
[8] J. Duchi, E. Hazan, and Y. Singer, *Adaptive subgradient methods for online learning and stochastic optimization*, Journal of Machine Learning Research, 12 (2011), pp. 2121–2159, http://jmlr.org/papers/v12/duchi11a.html.
[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, 2016.
[10] G. J. Gordon, *Regret bounds for prediction problems*, in Proceedings of the 12th Annual Conference on Computational Learning Theory (COLT), Santa Cruz, CA, USA, 1999, ACM, pp. 29–40.
[11] A. Granas and J. Dugundji, *Fixed Point Theory*, Springer Monographs in Mathematics, Springer, New York, 2003.
[12] G. Hinton, N. Srivastava, and K. Swersky, *Lecture 6a: Overview of mini-batch gradient descent*. https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf, 2012. Neural Networks for Machine Learning, University of Toronto.
[13] S. Ioffe and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, in Proceedings of the 32nd International Conference on Machine Learning (ICML), vol. 37 of Proceedings of Machine Learning Research, PMLR, 2015, pp. 448–456, https://proceedings.mlr.press/v37/ioffe15.html.
[14] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017, https://arxiv.org/abs/1412.6980, https://arxiv.org/abs/1412.6980.
[15] A. Krizhevsky, *Learning multiple layers of features from tiny images*, Tech. Report Technical Report, University of Toronto, 2009, https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf. Technical Report.
[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, in Advances in Neural Information Processing Systems, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds., vol. 25, Curran Associates, Inc., 2012, https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.
[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, in Advances in Neural Information Processing Systems, vol. 25, Curran Associates, Inc., 2012, https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.
[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278–2324, https://doi.org/10.1109/5.726791.
[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278–2324, https://doi.org/10.1109/5.726791.
[20] Y. LeCun, C. Cortes, and C. J. Burges, *MNIST handwritten digit database*. http://yann.lecun.com/exdb/mnist, 2010. Accessed: 2024-05-16.
[21] H. Li, A. Rakhlin, and A. Jadbabaie, *Convergence of Adam under relaxed assumptions*, in Advances in Neural Information Processing Systems, vol. 36, 2024. To appear.
[22] G. J. Minty, *Monotone (nonlinear) operators in Hilbert space*, Duke Mathematical Journal, 29 (1962), pp. 341–346, https://doi.org/10.1215/S0012-7094-62-02933-2.
[23] K. Nar and S. Sastry, *Step size matters in deep learning*, Advances in Neural Information Processing Systems, 31 (2018).
[24] S. J. Reddi, S. Kale, and S. Kumar, *On the convergence of adam and beyond*, in Proceedings of the 6th International Conference on Learning Representations (ICLR), 2018, https://openreview.net/forum?id=ryQu7f-RZ. Conference version of arXiv:1904.09237.
[25] H. Robbins and S. Monro, *A Stochastic Approximation method*, The Annals of Mathematical Statistics, 22 (1951), pp. 400–407, https://doi.org/10.1214/aoms/1177729586.
[26] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[27] S. Ruder, *An overview of gradient descent optimization algorithms*, 2016, https://doi.org/10.48550/arXiv.1609.04747, https://arxiv.org/abs/1609.04747, https://arxiv.org/abs/1609.04747. arXiv preprint arXiv:1609.04747.
[28] W. Rudin, *Principles of Mathematical Analysis*, McGraw-Hill, New York, 3rd ed., 1976.
[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning representations by Back-Propagating Errors*, Nature, 323 (1986), pp. 533–536, https://doi.org/10.1038/323533a0.
[30] S. Shalev-Shwartz, *Online learning and online convex optimization*, Foundations and Trends

in Machine Learning, 4 (2012), pp. 107–194, https://doi.org/10.1561/2200000018.
[31] N. Shi, D. Li, M. Hong, and R. Sun, *RMSprop converges with proper hyper-parameter*, in Proceedings of the International Conference on Learning Representations (ICLR), 2021, https://openreview.net/forum?id=3UDSdyIcBDA. OpenReview submission.
[32] W. A. Sutherland, *Introduction to Metric and Topological Spaces*, Oxford University Press, Oxford, 2nd ed., 2009.
[33] T. Tieleman, *Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude*, 2012, https://cir.nii.ac.jp/crid/1370017282431050757.
[34] L. Vandenberghe, *Optimization methods for large-scale systems lecture notes*, 2022, https://www.seas.ucla.edu/~vandenbe/236C/lectures/gradient.pdf. Lecture notes, UCLA.
[35] R. Ward, X. Wu, and L. Bottou, *AdaGrad stepsizes: Sharp convergence over nonconvex landscapes*, in Proceedings of the 36th International Conference on Machine Learning, K. Chaudhuri and R. Salakhutdinov, eds., vol. 97 of Proceedings of Machine Learning Research, PMLR, 09–15 Jun 2019, pp. 6677–6686, https://proceedings.mlr.press/v97/ward19a.html.
[36] H. Xiao, K. Rasul, and R. Vollgraf, *Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms*, 2017, https://arxiv.org/abs/1708.07747, https://arxiv.org/abs/1708.07747. arXiv preprint arXiv:1708.07747.
[37] M. D. Zeiler, *ADADELTA: An Adaptive Learning Rate Method*, arXiv e-prints, (2012), arXiv:1212.5701, p. arXiv:1212.5701, https://doi.org/10.48550/arXiv.1212.5701, https://arxiv.org/abs/1212.5701.
[38] Y. Zhang, C. Chen, N. Shi, R. Sun, and Z.-Q. Luo, *Adam can converge without any modification on update rules*, in Advances in Neural Information Processing Systems, vol. 35, 2022, pp. 28386–28399.
[39] M. Zinkevich, *Online convex programming and generalized infinitesimal gradient ascent*, in Proceedings of the 20th International Conference on Machine Learning (ICML), ICML'03, Washington, DC, USA, 2003, AAAI Press, pp. 928–935, https://dl.acm.org/doi/10.5555/3041838.3041955.

Appendix A. Deterministic Convergence Proof.

DEFINITION A.1. *Let $G : \mathbb{R}^n \to \mathbb{R}^n$ and let $\beta > 0$. The operator G is said to be* β-cocoercive *if*

$$\langle G(x) - G(y),\, x - y\rangle \geq \beta \,\|G(x) - G(y)\|^2 \quad \textit{for all } x, y \in \mathbb{R}^n. \tag{A.1}$$

Equivalently, for $L := 1/\beta > 0$, G is $(1/L)$-cocoercive *if*

$$\frac{1}{L}\,\|G(x) - G(y)\|^2 \leq \langle G(x) - G(y),\, x - y\rangle \quad \textit{for all } x, y \in \mathbb{R}^n. \tag{A.2}$$

When $G = \nabla f$ for a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, we say that f has β-cocoercive gradient *if ∇f is β-cocoercive in the sense of* (A.1).

LEMMA A.2. *Suppose ∇f is Lipschitz continuous with parameter L, domain of f is $\mathbb{R}^n$ and f has a minimum at x^*. Then*

$$\frac{1}{2L}\|\nabla f(z)\|^2 \leq f(z) - f(x^*).$$

Proof. The proof relies on the following quadratic upper bound property

$$f(y) \leq f(z) + \langle \nabla f(z), y - z\rangle + \frac{L}{2}\|y - z\|^2 \quad \text{for all } y, z \in \ \mathrm{dom}(f) \tag{A.3}$$

where $\mathrm{dom}(f)$ is a convex set. Taking infimum on both sides of A.3 gives

$$\begin{aligned} f(x^*) = \inf_y f(y) &\leq \inf_y \left(f(z) + \langle \nabla f(z), y - z\rangle + \frac{L}{2}\|y - z\|^2 \right) \\ &= \inf_{\|v\|=1} \inf_t \left(f(z) + t\langle \nabla f(z), v\rangle + \frac{Lt^2}{2} \right) \\ &= \inf_{\|v\|=1} \left(f(z) - \frac{1}{2L}\langle \nabla f(z), v\rangle \right) \\ &= f(z) - \frac{1}{2L}\|\nabla f(z)\|^2. \end{aligned} \tag{A.4}$$

Rearranging gives the claim. ∎

LEMMA A.3 (Cocoercivity). *Assume f is convex, proper and lower semicontinuous. Let ∇f be Lipschitz and let L_G be the Lipschitz constant for $\dfrac{\nabla f}{\sqrt{\|\nabla f\|^2 + 1}}$. Then the following cocoercivity property holds:*

$$\begin{aligned} &\frac{1}{L_G}\left\| \frac{\nabla f(x)}{\sqrt{\|\nabla f(x)\|^2 + 1}} - \frac{\nabla f(y)}{\sqrt{\|\nabla f(y)\|^2 + 1}} \right\|^2 \\ &\qquad \leq \left\langle \frac{\nabla f(x)}{\sqrt{\|\nabla f(x)\|^2 + 1}} - \frac{\nabla f(y)}{\sqrt{\|\nabla f(y)\|^2 + 1}}, x - y \right\rangle \end{aligned}$$

Proof. For any x and y, let

$$\begin{aligned} f_x(z) &\coloneqq F(z) - \frac{\nabla f(x)^T z}{\sqrt{\|\nabla f(x)\|^2 + 1}}, \\ f_y(z) &\coloneqq F(z) - \frac{\nabla f(y)^T z}{\sqrt{\|\nabla f(y)\|^2 + 1}} \end{aligned}$$

for some constant a where $F(z) \coloneqq \int_a^z \frac{\nabla f(u)}{\sqrt{\|\nabla f(u)\|^2+1}} du$ is a scalar potential function for the vector field integrand. Notice that F is well defined since the integrand is a conservative vector field. Also, $F(z)$ is convex since the integrand is a maximal monotone operator resulting from a convex, proper and lower semi-continuous function, f [22, 4, 26].

Thus, f_x and f_y are well defined and convex since the difference of a convex function and a linear function is convex. f_x is minimized at $z = x$, thus evaluating f_x at y and x and subtracting the results gives

$$\begin{aligned} & F(y) - F(x) - \left\langle \frac{\nabla f(x)}{\sqrt{\|\nabla f(x)\|^2+1}}, y - x \right\rangle \\ &= f_x(y) - f_x(x) \\ &\geq \frac{1}{2L_G} \|\nabla f_x(y)\|^2 \qquad \text{(A.5)} \\ &= \frac{1}{2L_G} \left\| \frac{\nabla f(y)}{\sqrt{\|\nabla f(y)\|^2+1}} - \frac{\nabla f(x)}{\sqrt{\|\nabla f(x)\|^2+1}} \right\|^2. \qquad \text{(A.6)} \end{aligned}$$

Similarly, $z = y$ minimizes f_y and

$$\begin{aligned} & F(x) - F(y) - \left\langle \frac{\nabla f(y)}{\sqrt{\|\nabla f(y)\|^2+1}}, x - y \right\rangle \\ &= f_y(x) - f_y(y) \\ &\geq \frac{1}{2L_G} \|\nabla f_y(x)\|^2 \\ &= \frac{1}{2L_G} \left\| \frac{\nabla f(y)}{\sqrt{\|\nabla f(y)\|^2+1}} - \frac{\nabla f(x)}{\sqrt{\|\nabla f(x)\|^2+1}} \right\|^2. \qquad \text{(A.7)} \end{aligned}$$

Adding A.6 and A.7 yields the result. ∎

LEMMA A.4 (Existence of Fix(T)). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be proper, lower semicontinuous and convex. If f is coercive, i.e.,*

$$\|x\| \to \infty \quad \textit{implies} \quad f(x) \to \infty,$$

then f attains a minimizer $x^* \in \arg\min_x f(x)$ and $\arg\min_x f \neq \emptyset$. If in addition f is differentiable, then

$$\arg\min_x f = \{x \in \mathbb{R}^n : \nabla f(x) = 0\}.$$

Consequently, for the operator T defined in (5.2)*,*

$$\mathrm{Fix}(T) = \arg\min_x f \neq \emptyset.$$

Proof. Existence of a minimizer for proper, lower semicontinuous, coercive convex functions on $\mathbb{R}^n$ is standard; see, e.g., [1, 26]. For differentiable convex f, first-order optimality gives $\nabla f(x) = 0$ if and only if $x \in \arg\min_x f$ (see, e.g., [1, Proposition 17.6]). Finally, since

$$T(x) = x - \gamma \frac{\nabla f(x)}{\sqrt{\|\nabla f(x)\|^2+1}},$$

we have $T(x) = x$ if and only if $\nabla f(x) = 0$, hence $\mathrm{Fix}(T) = \arg\min_x f$. ∎

THEOREM A.5 (Deterministic convergence). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and continuously differentiable. Assume in addition that f is proper and lower semicontinuous, and that f is coercive. Let*

$$g(x) \coloneqq \frac{\nabla f(x)}{\sqrt{\|\nabla f(x)\|^2 + 1}} \qquad \text{and} \qquad T(x) \coloneqq (\mathrm{I} - \gamma g)(x),$$

where I *is the identity map and $L_G > 0$ denotes the Lipschitz constant for g. Then f attains a minimizer $x^* \in \arg\min f$, and $\mathrm{Fix}(T) = \arg\min_x f \neq \emptyset$. If $0 < \gamma < \frac{2}{L_G}$, then T is nonexpansive,*

$$\|Tx - Ty\| \leq \|x - y\| \quad \text{for all } x, y \in \mathbb{R}^n.$$

Moreover, for the fixed-point iteration $x_{k+1} = T(x_k)$, the sequence $\{x_k\}$ converges to a point $x^\star \in \mathrm{Fix}(T) = \arg\min_x f$.

Proof. First, we compute as follows

$$\begin{aligned}
\|Tx - Ty\|^2 &= \left\| x - \frac{\gamma \nabla f(x)}{\sqrt{\|\nabla f(x)\|^2 + 1}} - y + \frac{\gamma \nabla f(y)}{\sqrt{\|\nabla f(y)\|^2 + 1}} \right\|^2 \\
&= \left\| (x - y) - \gamma \left(\frac{\nabla f(x)}{\sqrt{\|\nabla f(x)\|^2 + 1}} - \frac{\nabla f(y)}{\sqrt{\|\nabla f(y)\|^2 + 1}} \right) \right\|^2 \\
&= \|x - y\|^2 - 2\gamma \left\langle x - y, \frac{\nabla f(x)}{\sqrt{\|\nabla f(x)\|^2 + 1}} - \frac{\nabla f(y)}{\sqrt{\|\nabla f(y)\|^2 + 1}} \right\rangle \\
&\quad + \gamma^2 \left\| \frac{\nabla f(x)}{\sqrt{\|\nabla f(x)\|^2 + 1}} - \frac{\nabla f(y)}{\sqrt{\|\nabla f(y)\|^2 + 1}} \right\|^2 \\
&\leq \|x - y\|^2 + \left(\gamma^2 - \frac{2\gamma}{L_G} \right) \left\| \frac{\nabla f(x)}{\sqrt{\|\nabla f(x)\|^2 + 1}} - \frac{\nabla f(y)}{\sqrt{\|\nabla f(y)\|^2 + 1}} \right\|^2 \qquad \text{(A.8)} \\
&= \|x - y\|^2 - \gamma \left(\frac{2}{L_G} - \gamma \right) \left\| \frac{\nabla f(x)}{\sqrt{\|\nabla f(x)\|^2 + 1}} - \frac{\nabla f(y)}{\sqrt{\|\nabla f(y)\|^2 + 1}} \right\|^2. \qquad \text{(A.9)}
\end{aligned}$$

Inequality (A.8) follows from cocoercivity (Lemma A.3). If $0 < \gamma < 2/L_G$, then (A.9) implies

$$\|Tx - Ty\|^2 \leq \|x - y\|^2,$$

hence T is nonexpansive.
By Lemma A.4, f attains a minimizer $x^\star \in \arg\min_x f$. Hence, $\mathrm{Fix}(T) = \arg\min_x f \neq \emptyset$. Choose $x^\star \in \mathrm{Fix}(T)$ and take $y = x^\star$ in (A.9). Since $T(x^\star) = x^\star$, we have $\nabla f(x^\star) = 0$ and thus

$$\|x_{k+1} - x^\star\|^2 \leq \|x_k - x^\star\|^2 - \gamma \left(\frac{2}{L_G} - \gamma \right) \left\| \frac{\nabla f(x_k)}{\sqrt{\|\nabla f(x_k)\|^2 + 1}} \right\|^2. \qquad \text{(A.10)}$$

□

Thus $\{\|x_k - x^\star\|\}$ is nonincreasing and $\{x_k\}$ is bounded.
Summing (A.10) from $k = 0$ to N yields

$$\gamma \left(\frac{2}{L_G} - \gamma \right) \sum_{k=0}^{N} \left\| \frac{\nabla f(x_k)}{\sqrt{\|\nabla f(x_k)\|^2 + 1}} \right\|^2 \leq \|x_0 - x^\star\|^2,$$

so

$$\sum_{k=0}^{\infty} \left\| \frac{\nabla f(x_k)}{\sqrt{\|\nabla f(x_k)\|^2 + 1}} \right\|^2 < \infty.$$

In particular,

$$\left\| \frac{\nabla f(x)}{\sqrt{\|\nabla f(x)\|^2 + 1}} \right\| \quad \text{vanishes iff} \quad \|\nabla f(x)\| = 0$$

Since $\|x_{k+1} - x_k\| = \gamma \left\| \frac{\nabla f(x_k)}{\sqrt{\|\nabla f(x_k)\|^2+1}} \right\|$, it follows that $\|x_{k+1} - x_k\| \to 0$ as $k \to \infty$. By boundedness of $\{x_k\}$, the Bolzano-Weierstrass theorem implies existence of a convergent subsequence, namely there exist $\{k_j\}$ and $\bar{x} \in \mathbb{R}^n$ such that $x_{k_j} \to \bar{x}$ [1, 7]. By continuity of ∇f and $\nabla f(x_{k_j}) \to 0$, we obtain $\nabla f(\bar{x}) = 0$, hence $\bar{x} \in \text{Fix}(T)$.

Finally, (A.10) implies that for this $\bar{x}$ the sequence $\|x_k - \bar{x}\|^2$ is nonincreasing and has a subsequence converging to 0. Hence $\|x_k - \bar{x}\|^2 \to 0$. Therefore $x_k \to \bar{x} \in \text{Fix}(T)$.

A best-iterate residual rate follows directly from the telescoping bound. For any $N \geq 0$,

$$\min_{0 \leq k \leq N} \left\| \frac{\nabla f(x_k)}{\sqrt{\|\nabla f(x_k)\|^2 + 1}} \right\|^2 \leq \frac{\|x_0 - x^\star\|^2}{\gamma(2/L_G - \gamma)(N+1)}.$$

Appendix B. Stochastic Convergence Proof.

B.1. Important Lemmas and Definitions. In this section, we provide proof of convergence of our algorithm. To this end, we start with some definitions and lemmas necessary for the main theorem.

DEFINITION B.1. *A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if for all $x, y \in \mathbb{R}$,*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x). \tag{B.1}$$

For the rest of the paper, we make the following assumptions on the stochastic objective function $f_t : \mathbb{R}^n \to \mathbb{R}$ where the iteration counter $t \geq 1$.

ASSUMPTIONS B.2.
1. *f_t is convex for all t.*
2. *For all t, f_t is differentiable.*
3. *For all t, there exists $G \geq 0$ such that $\|\nabla f_t(x)\| \leq G$ for all $x \in X \subseteq \mathbb{R}^n$ where X is the feasible set.*
4. *$A \coloneqq \{x_1, x_2, \cdots\}$, the iterates generated by GeoAdaLer algorithm.*
5. *$x^* \coloneqq \arg\min_{x \in X} \sum_{t=1}^{T} f_t(x)$ exists and $\|x_k - x^*\| \leq D$ for all $x_k \in A$.*
6. *$\beta_t \coloneqq \beta \lambda^{t-1}$ where $\lambda \in (0, 1), \beta \in [0, 1)$.*

Again, for notational convenience, we take $g_t = \nabla f_t(x_t)$.

LEMMA B.3. *Under Assumptions B.2, no. 3, the exponential moving average*

$$m_t = \beta_t m_{t-1} + (1 - \beta_t) g_t \tag{B.2}$$

is bounded for all t.

Proof. Iteratively expanding out m_t, we obtain

$$m_t = (1-\beta_t)\sum_{i=1}^{t}\beta_t^{t-i+1}g_i.$$

Taking the Euclidean norm on both sides and using the boundedness of g_t gives

$$\begin{aligned}\|m_t\| &= \|(1-\beta_t)\sum_{i=1}^{t}\beta_t^{t-i+1}g_i\| \\ &\leq (1-\beta_t)\sum_{i=1}^{t}\beta_t^{t-i+1}\|g_i\| \\ &\leq (1-\beta_t)G\sum_{i=1}^{t}\beta_t^{t-i+1} \\ &= G(1-\beta_t^t) \leq G\end{aligned}$$

∎

LEMMA B.4. *Under Assumptions B.2, no. 6, the following inequalities hold*

1. $\frac{\beta_t}{1-\beta_t} \leq \frac{\beta}{1-\beta}$
2. $\frac{1}{1-\beta_t} \leq \frac{1}{1-\beta}$
3. $\sum_{t=1}^{T}\frac{\beta_t}{1-\beta_t} \leq \frac{\beta(1-\lambda^T)}{(1-\beta)(1-\lambda)}$

for all t.

Proof. 1). For all $t, \lambda < 1$ gives $\lambda^{t-1} < 1$ so that $\frac{1}{\lambda^{1-t}} < 1$. Thus

$$\frac{\beta_t}{1-\beta_t} = \frac{\beta\lambda^{t-1}}{1-\beta\lambda^{t-1}} = \frac{\beta}{\lambda^{1-t}-\beta} \leq \frac{\beta}{1-\beta}.$$

2). $\beta_t = \beta\lambda^{t-1} \leq \beta$ for all t since $\lambda \in (0,1), \beta \in [0,1)$. So, $1-\beta \leq 1-\beta\lambda^{t-1}$ leads to

$$\frac{1}{1-\beta_t} \leq \frac{1}{1-\beta},$$

as required.

3). By Lemma B.4 no. 2, we have

$$\sum_{t=1}^{T}\frac{\beta_t}{1-\beta_t} \leq \sum_{t=1}^{T}\frac{\beta_t}{1-\beta} = \frac{\beta}{1-\beta}\sum_{t=1}^{T}\lambda^{t-1} = \frac{\beta(1-\lambda^T)}{(1-\beta)(1-\lambda)}.$$

∎

LEMMA B.5. *For all $t \geq 1$ and for all $T \geq t$, the following inequality holds*

$$\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \leq 1 + \int_1^T \frac{1}{\sqrt{t}}dt. \tag{B.3}$$

Proof trivially follows from the integral test for convergence of series and is also given in [2].

LEMMA B.6. *Under Assumptions B.2, no. 1 and 2, the following inequality holds for all t*

$$R(T) \leq \sum_{t=1}^{T} g_t \cdot (x_t - x^*). \tag{B.4}$$

Proof. By converxity of f_t for each t,

$$f_t(x^*) - f_t(x_t) \geq \nabla f_t(x_t) \cdot (x^* - x_t).$$

It then follows that

$$f_t(x_t) - f_t(x^*) \leq \nabla f_t(x_t) \cdot (x_t - x^*) = g_t \cdot (x_t - x^*).$$

Hence, summing both sides from $t = 1$ to T gives the required result. ∎

Appendix C. Proof of Theorem 5.2. From the update rule in Algorithm 3.1

$$x_{t+1} = x_t - \gamma_t \frac{m_t}{\sqrt{\|m_t\|^2 + 1}}.$$

Subtracting x^* from both sides and taking norm squared, we obtain

$$\begin{aligned} \|x_{t+1} - x^*\|^2 &= |x_t - \gamma_t \frac{m_t}{\sqrt{\|m_t\|^2 + 1}}|^2 \\ &= \|x_t - x^*\|^2 - \frac{2\gamma_t}{\sqrt{\|m_t\|^2 + 1}} m_t \cdot (x_t - x^*) + \gamma_t^2 \frac{\|m_t\|^2}{\|m_t\|^2 + 1}. \end{aligned}$$

Substituting $m_t = \beta_t m_{t-1} + (1 - \beta_t) g_t$, we obtain

$$\begin{aligned} \|x_{t+1} - x^*\|^2 = \|x_t - x^*\|^2 &- \frac{2\gamma_t}{\sqrt{\|m_t\|^2 + 1}} m_{t-1} \cdot (x_t - x^*) \\ &- \frac{2\gamma_t(1 - \beta_t)}{\sqrt{\|g_t\|^2 + 1}} g_t \cdot (x_t - x^*) + \gamma_t^2 \frac{\|m_t\|^2}{\|m_t\|^2 + 1}. \end{aligned}$$

Rearrange to have $g_t \cdot (x_t - x^*)$ on the left hand side:

$$\begin{aligned} g_t \cdot (x_t - x^*) &= \frac{\sqrt{\|m_t\|^2 + 1}}{2\gamma_t(1 - \beta_t)} \left[\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2\right] - \frac{\beta_t}{1 - \beta_t} m_{t-1} \cdot (x_t - x^*) \\ &\quad + \frac{\gamma_t}{2(1 - \beta_t)} \frac{\|m_t\|^2}{\sqrt{\|m_t\|^2 + 1}} \\ &\leq \frac{\sqrt{\|m_t\|^2 + 1}}{2\gamma_t(1 - \beta_t)} \left[\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2\right] \\ &\quad + \frac{\beta_t}{1 - \beta_t} \|m_{t-1}\| \|(x^* - x_t)\| + \frac{\gamma_t \|m_t\|^2}{2(1 - \beta_t)}. \end{aligned} \tag{C.1}$$

where inequality C.1 follows from the Cauchy-Schwartz inequality applied to the second term $-m_{t-1} \cdot (x_t - x^*) = m_{t-1} \cdot (x^* - x_t)$ and the fact that $\frac{\|m_t\|^2}{\sqrt{\|m_t\|^2 + 1}} \leq 1$ for all t.

Summing both sides from $t = 1$ to T and using Lemma B.3 and Assumptions B.2 no. 3 and 5, we further obtain

$$\begin{aligned}\sum_{t=1}^{T} g_t \cdot (x_t - x^*) &\leq \sum_{t=1}^{T} \frac{\sqrt{G^2+1}}{2\gamma_t(1-\beta_t)} \left[\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2\right] \\ &\quad + DG \sum_{t=1}^{T} \frac{\beta_t}{1-\beta_t} + G^2 \sum_{t=1}^{T} \frac{\gamma_t}{2(1-\beta_t)} \\ &\leq \sum_{t=1}^{T} \frac{\sqrt{G^2+1}}{2\gamma_t(1-\beta_t)} \left[\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2\right] \\ &\quad + \frac{DG\beta(1-\lambda^T)}{(1-\beta)(1-\lambda)} + \frac{G^2}{2(1-\beta)} \sum_{t=1}^{T} \gamma_t. \end{aligned} \tag{C.2}$$

where the last inequality follows from Assumptions B.2 no. 3 and Lemma B.4 no. 2 and 3.
By expanding the first term and and rewriting the what is left in compact form, we obtain

$$\begin{aligned}&\sum_{t=1}^{T} \frac{1}{\gamma_t(1-\beta_t)} \left[\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2\right] \\ &\quad = \frac{1}{\gamma_1(1-\beta_1)} \|x_1 - x^*\|^2 - \frac{1}{\gamma_T(1-\beta_T)} \|x_{T+1} - x^*\|^2 \\ &\qquad + \sum_{t=2}^{T} \left(\frac{1}{\gamma_t(1-\beta_t)} - \frac{1}{\gamma_{t-1}(1-\beta_{t-1})} \right) \|x_t - x^*\|^2 \\ &\quad \leq \frac{\|x_1 - x^*\|^2}{\gamma_1(1-\beta_1)} \\ &\qquad + \sum_{t=1}^{T} \left(\frac{1}{\gamma_t(1-\beta_t)} - \frac{1}{\gamma_{t-1}(1-\beta_{t-1})} \right) \|x_t - x^*\|^2. \end{aligned}$$

where the last inequality follows from dropping the second term. By the Assumptions B.2 no. 5, it further simplifies to

$$\begin{aligned}&\sum_{t=1}^{T} \frac{1}{\gamma_t(1-\beta_t)} \left[\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2\right] \\ &\quad \leq \frac{D^2}{\gamma_1(1-\beta_1)} + D^2 \sum_{t=2}^{T} \left(\frac{1}{\gamma_t(1-\beta_t)} - \frac{1}{\gamma_{t-1}(1-\beta_{t-1})} \right) \\ &\quad = \frac{D^2}{\gamma_1(1-\beta_1)} - \frac{D^2}{\gamma_1(1-\beta_1)} + \frac{D^2}{\gamma_T(1-\beta_T)} \\ &\quad = \frac{D^2}{\gamma_T(1-\beta_T)}. \end{aligned}$$

From Lemma B.4 we have.

$$\sum_{t=1}^{T} \frac{1}{\gamma_t(1-\beta_t)} \left[\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2\right] \leq \frac{D^2}{\gamma_T(1-\beta)}.$$

Therefore equation C.2 become:

$$\sum_{t=1}^{T} g_t \cdot (x_t - x^*) \leq \frac{D^2\sqrt{G^2+1}}{\gamma_T(1-\beta_T)} + \frac{DG\beta(1-\lambda^T)}{(1-\beta)(1-\lambda)} + \frac{G^2}{2(1-\beta)} \sum_{t=1}^{T} \gamma_t. \tag{C.3}$$

Assuming $\gamma_t = \frac{1}{\sqrt{t}}$ we further obtain:

$$\sum_{t=1}^{T} g_t \cdot (x_t - x^*) \leq \frac{\sqrt{T}D^2\sqrt{G^2+1}}{(1-\beta_T)} + \frac{DG\beta(1-\lambda^T)}{(1-\beta)(1-\lambda)} + \frac{G^2}{2(1-\beta)} \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \tag{C.4}$$

$$\leq \frac{\sqrt{T}D^2\sqrt{G^2+1}}{(1-\beta_T)} + \frac{DG\beta(1-\lambda^T)}{(1-\beta)(1-\lambda)} + \frac{G^2(2T-1)}{2(1-\beta)}. \tag{C.5}$$

Equation C.5 follows from Lemma B.5 and so our regret is bounded above by:

$$R(T) \leq \frac{\sqrt{T}D^2\sqrt{G^2+1}}{(1-\beta_T)} + \frac{DG\beta(1-\lambda^T)}{(1-\beta)(1-\lambda)} + \frac{G^2(2T-1)}{2(1-\beta)}.$$

Appendix D. Datasets . MNIST: The MNIST database of handwritten digits. Licensed under the Creative Commons Attribution 4.0 License[20].

CIFAR-10: The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. Licensed under the Creative Commons Attribution 4.0 License [15].

Fashion MNIST: The Fashion MNIST database of fashion images. Licensed under The MIT License (MIT).[36]