AEGD: ADAPTIVE GRADIENT DESCENT WITH ENERGY*

HAILIANG LIU† AND XUPING TIAN‡

Abstract. We propose AEGD, a new algorithm for optimization of non-convex objective functions, based on a dynamically updated 'energy' variable. The method is shown to be unconditionally energy stable, irrespective of the base step size. We prove energy-dependent convergence rates of AEGD for both non-convex and convex objectives, which for a suitably small step size recovers desired convergence rates for the batch gradient descent. We also provide an energy-dependent bound on the stationary convergence of AEGD in the stochastic non-convex setting. The method is straightforward to implement and requires little tuning of hyper-parameters. Experimental results demonstrate that AEGD works well for a large variety of optimization problems. Specifically, it is robust with respect to initial data, capable of making rapid initial progress. The stochastic AEGD shows comparable and often better generalization performance than SGD with momentum for deep neural networks. The code is available at https://github.com/txping/AEGD.

Key words. Stochastic optimization, gradient descent, energy stability

AMS subject classifications. 90C15, 65K10, 68Q25

1. Introduction.

From a mathematical perspective, training neural networks is a high-dimensional non-convex optimization problem, and the dynamics of a training process can be incredibly complicated. Despite this, stochastic gradient descent (SGD) [48] methods have proven to be extremely effective for training neural networks in practice (see, for example, [11, 18]). Such stochastic training often reduces to solving the following unconstrained minimization problem

$$\min_{\theta\in\mathbb{R}^n} f(\theta), \tag{1.1}$$

where $f:\mathbb{R}^n\to\mathbb{R}$ is a finite-sum function, defined as $f(\theta):=\frac{1}{m}\sum_{i=1}^m f_i(\theta)$, where $f_i(\theta):=l(F(x_i;\theta);y_i)$ is a loss of the machine learning (ML) model $F(\cdot;\theta)$ on the training data $\{x_i,y_i\}$, parametrized by θ. For many practical applications, $f(\theta)$ is highly non-convex, and $F(\cdot;\theta)$ is chosen among deep neural networks (DNNs), known for their superior performance across various tasks. These DNNs are heavily over-parametrized and require large amounts of training data. Thus, both m and n can scale up to millions or even billions. These complications pose serious computational challenges.

Gradient descent (GD) or its variants are methods of choice for solving (1.1). There are two variations of gradient decent, which differ in how much data we use to compute the gradient of the objective function. They are fully gradient decent, aka batch GD, and SGD. For large dataset, SGD is usually much faster by performing a parameter update for each training example. The GD method is implemented as the recursive rule given an initial point θ_0,

$$\theta_{k+1}=\theta_k-\eta\nabla f(\theta_k), \tag{1.2}$$

where $\eta>0$ is the step size (called the learning rate in ML). GD has advantages of easy implementation and being fast for well-conditioned and strongly convex objectives,

* **Funding:** We would like to acknowledge support for this project from the National Science Foundation (NSF grant DMS-1812666).

†Department of Mathematics, Iowa State University, Ames, IA 50011, USA (hliu@iastate.edu, https://faculty.sites.iastate.edu/hliu/).

‡Department of Mathematics, Iowa State University, Ames, IA 50011, USA (xupingt@iastate.edu).

independent of the dimension of the underlying problem [41]. However, GD is stable typically when step size η is suitably small, hence slow down the convergence. Finding a good step size is an important practical problem [14, 41]. The step limitation issue can be largely resolved by implicit updates,

$$\theta_{k+1} = \theta_k - \eta \nabla f(\theta_{k+1}). \tag{1.3}$$

This has an intimate relationship with the proximal point algorithm (PPA) [49]:

$$\theta_{k+1} = \text{argmin}_{\theta \in \mathbb{R}^n} \left\{ f(\theta) + \frac{1}{2\eta} \|\theta - \theta_k\|^2 \right\}. \tag{1.4}$$

The advantage of this method is that the objective function value $f(\theta_k)$ decreases monotonically for any step size $\eta > 0$. However, for non-quadratic functions f, one would have to solve it by an internal iteration at each step. For example, by the backward Euler method [58]. A natural question is:

Can we adapt (1.2) so that it becomes stable for arbitrary step size as in (1.3) but still easy to implement?

We answer this question by proposing AEGD, a new method for efficient optimization that only requires first-order gradients. The key idea with AEGD is to introduce $g = \sqrt{f + c}$ so that $f + c > 0$, and split the gradient $\nabla f(\theta)$ as a product $\nabla f(\theta) = 2g(\theta)\nabla g(\theta)$. We then apply an implicit-explicit selection to the product at $k-$th iteration to obtain

$$\theta_{k+1} = \theta_k - 2\eta r_{k+1} \nabla g(\theta_k),$$

while r_k also updates, according to $r_{k+1} - r_k = \nabla g(\theta_k) \cdot (\theta_{k+1} - \theta_k)$. This allows us to obtain a base formulation:

$$r_{k+1} = \frac{r_k}{1 + 2\eta |v_k|^2},$$
$$\theta_{k+1} = \theta_k - 2\eta r_{k+1} v_k,$$

where $v_k = \nabla g(\theta_k)$ or an unbiased estimate of $\nabla g(\theta_k)$ for stochastic AEGD. Two distinct features with this method are: (i) it is easy to implement, since only explicit updates are involved; (ii) it is unconditionally energy stable in the sense that r_k as an approximation of $g(\theta_k)$ is a decreasing sequence in k, for any step size $\eta > 0$.

To the best of our knowledge the idea of using energy update to stabilize GD in the optimization of non-convex objective functions was not explored earlier. Meanwhile we show that AEGD feathers the expected convergence rates for first-order gradient methods. At the core of AEGD is the auxiliary energy variable which serves to adjust the effective step size at each update. The name AEGD is derived from the adaptive GD method with energy. We would like to emphasize that our AEGD features both a global and an element-wise formulation, while by empirical evidence, the latter appears to perform better in most cases.

1.1. Main Contributions. Some major benefits of AEGD are summarized below:

- It is shown to be unconditionally energy stable, irrespective of the base step size.
- It allows a larger base step size than GD.
- It converges fast even for objective functions that have a large condition number.

- It is robust with respect to initial data.
- It has visible advantages over the GD algorithm with momentum.

Most of the above advantages are kept for the stochastic AEGD, which will be discussed in Section 4. In particular, our numerical results also show that the stochastic AEGD has the potential for training neural networks with better generalization, at least on some benchmark machine learning tasks.

1.2. Further Related Work. A numerical method bearing a direct relation to AEGD is the invariant energy quadratization (IEQ) approach introduced by [57] and [62] for gradient flows in the form of partial differential equations (PDEs). The IEQ approach uses a modified quadratic energy to construct linear, unconditionally energy stable schemes for solving time-dependent PDEs [31, 32, 53]. The AEGD algorithm here is new in that we use an auxiliary energy variable to obtain unconditionally energy stable optimization algorithms.

In the field of stochastic optimization, there is an extensive volume of research for designing algorithms to speed up the convergence of SGD. Among these, main advances fall into two categories: momentum methods [40, 44] and adaptive step size methods [15, 55, 25]. The integration of momentum and adaptive step size led to Adam [25], which is one of the optimization algorithms that are widely used by the deep learning community. Many further advances have improved Adam [50, 35, 36].

The name momentum stems from an analogy to momentum in physics. GD with constant momentum such as the heavy-ball (HB) method [44] is known to enjoy the convergence rate of $O(1/k)$ for convex smooth optimization. Remarkably, with an adaptive momentum the Nesterov accelerated gradient (NAG) [40, 54] has the convergence rate up to the optimal $O(1/k^2)$. Recent advances showed that NAG has other advantages such as speeding up escaping saddle points [22], accelerating SGD or GD in non-convex problems [47, 63]. Both first and second order optimization methods can be designed via inertial systems [3, 12].

Another major bottleneck for fast convergence of SGD lies in its variance [9, 51, 52]. Different ideas have been proposed to develop variance reduction algorithms. Some representative works are SAGA [13], SCSG [28], SVRG [23], Laplacian smoothing [43, 56], and iterative averaging methods [45, 61].

For a gradient based method the geometric property of f often affects the convergence rates. For the non-convex f we consider an old condition originally introduced by Polyak [46], who showed that it is a sufficient condition for gradient descent to achieve a linear convergence rate; see [24] for a recent convergence study under this condition. Such condition is often called the Polyak-Łojasiewicz (PL) inequality in the literature since it is a special case of the inequality introduced by Łojasiewicz in 1963 [34]. A generalization of the PL property for non-smooth optimization is the KL inequality [27, 6]. The KL inequality has been successfully used to study the convergence of the proximal algorithms [2], and the proximal alternating and projection methods [1].

Finally, we mention that a considerable body of work exists concerning the related but distinct strategy of improving GD for unconstrained optimization problems. For line search based GD methods we recall the classical Armijo rule (1966) and the Wolfe conditions (1969) to guide the inexact line search, also referring to [60, 39], the book of [41], and the references therein. Another notable gradient method with modified step sizes is the BB method [4], which is motivated by Newton's method but not involves any Hessian. There are also studies of other GD-type methods that allow to handle constraints and non-smooth objective functions, and the possibility of speeding up [5, 16, 42, 29]; we refer the reader to [38, 17, 7] for work on minimizing composite

objectives, and also further references.

Even though we do not consider alternative methods or related extensions here, our present results for unconstrained optimization problems provide some understanding of the energy-driven update rule, which serves to create a path towards more alternative algorithms with energy.

1.3. Organization. In Section 2, we motivate and present the AEGD method for non-convex optimization in both global and element-wise version. Theoretical results, including stability, convergence, and convergence rates, are given in Section 3, with technical proofs deferred to Appendix A. In Section 4 we present the stochastic AEGD and some theoretical results. Section 5 provides experimental results to show the performance of AEGD. We end this paper with concluding remarks in Section 6.

1.4. Notation. Throughout this paper, we denote $\{1, \cdots, m\}$ by $[m]$ for integer m. For vectors and matrices we use $\|\cdot\|$ to denote the l_2-norm. For a function $f : \mathbb{R}^n \to \mathbb{R}$, we use ∇f and $\nabla^2 f$ to denote its gradient and Hessian, and θ^* to denote a local minimizer of f. We also use the notation $\partial_i := \partial_{\theta_i}$. In the algorithm description, we use $z = x/y$ to denote element-wise division if x and y are both vectors of size n; $x \odot y$ is element-wise product, and $x^2 = x \odot x$; $\mathbf{1} = (1, ..., 1)$ is a vector of all ones.

2. AEGD. Motivated by the invariant energy quadratizaton (IEQ) strategy, introduced in [57, 62] for discretization of PDEs, we define $g(\theta) = \sqrt{f(\theta) + c}$ with c chosen so that $f(\theta) + c > 0$. Set $r = g(\theta)$, then r^2 plays the role of energy and

$$\nabla f = 2r\nabla g, \quad r = g(\theta).$$

This transformation when applied to the usual gradient flow $\dot{\theta} = -\nabla f(\theta)$ leads to an augmented system

$$\dot{\theta} = -2r\nabla g(\theta), \tag{2.1a}$$

$$\dot{r} = \nabla g(\theta) \cdot \dot{\theta}. \tag{2.1b}$$

It allows the dissipation of the quadratic energy $\frac{d}{dt}(r^2) = -|\dot{\theta}|^2$ in place of the dissipation of nonlinear loss function: $\frac{d}{dt} f(\theta) = -|\dot{\theta}|^2$. Based on (2.1) we introduce the following update rule:

$$\theta_{k+1} = \theta_k - 2\eta r_{k+1}\nabla g(\theta_k), \quad k = 0, 1, 2, \cdots, \tag{2.2a}$$

$$r_{k+1} - r_k = \nabla g(\theta_k) \cdot (\theta_{k+1} - \theta_k). \tag{2.2b}$$

This is actually a linear algorithm with

$$r_{k+1} = \frac{r_k}{1 + 2\eta|\nabla g(\theta_k)|^2}, \quad r_0 = \sqrt{f(\theta_0) + c}, \tag{2.3a}$$

$$\theta_{k+1} = \theta_k - 2\eta r_{k+1}\nabla g(\theta_k), \quad k = 0, 1, 2, \cdots, \tag{2.3b}$$

and easy to implement. In order to allow the use of different step size in each coordinate, we also propose an element-wise AEGD, set forth as follows:

$$r_{k+1,i} = \frac{r_{k,i}}{1 + 2\eta(\partial_i g(\theta_k))^2}, \quad r_{0,i} = \sqrt{f(\theta_0) + c}, \quad i \in [n], \tag{2.4a}$$

$$\theta_{k+1,i} = \theta_{k,i} - 2\eta r_{k+1,i}\partial_i g(\theta_k), \quad k = 0, 1, 2, \cdots. \tag{2.4b}$$

This also implies

$$r_{k+1,i} - r_{k,i} = \partial_i g(\theta_k)(\theta_{k+1,i} - \theta_{k,i}), \quad i \in [n]. \tag{2.5}$$

Remark 2.1. One may wonder whether $r = (f + c)^\alpha$ is admissible for building similar algorithms for all $\alpha \in (0, 1)$. To see this, we write the corresponding augmented system as

$$\dot{\theta} = -\alpha^{-1} r^{1/\alpha - 1} \nabla g(\theta), \quad \dot{r} = \nabla g \cdot \dot{\theta}.$$

A similar implicit-explicit discretization can yield a linear update for r if and only if $\frac{1}{\alpha} - 1 = 1 \Leftrightarrow \alpha = \frac{1}{2}$. This explains why energy quadratization has been used in [57, 62] to name such strategy.

Remark 2.2. This method can also be linked with inertial methods at the continuous level. More precisely, from (2.1) we can derive a second order equation governed by

$$\ddot{\theta}(t) + 2\eta(\frac{d}{dt} g(\theta(t)))\nabla g(\theta(t)) + 2\eta g(\theta(t))\nabla^2 g(\theta(t))\dot{\theta}(t) = 0, \tag{2.6}$$

which belongs to the class of form

$$\ddot{\theta}(t) + \beta(t)\nabla^2 g(\theta(t))\dot{\theta}(t) + b(t)\nabla g(\theta(t)) = 0.$$

Related systems have been recently investigated in [3]. From their results, we can identify the strong aspect of AEGD: it is autonomous (adaptive), and involves strong damping governed by the Hessian of the function to be minimized. The method stability could be further improved if there is a way to incorporate an additional damping term in (2.6); in particular for functions which are not well conditioned.

3. Theoretical Results. In this section, we will show AEGD is unconditionally energy stable, irrespective of the step size η, and obtain different convergence rates for non-convex and convex functions. We say f is μ-strongly convex if for all u and v we have $f(u) \geq f(v) + \langle \nabla f(v), u - v\rangle + \mu\|u - v\|^2/2$; f is L-smooth if $\|\nabla^2 f(u)\| \leq L$ for any $u \in \mathbb{R}^n$. For convex ($\mu = 0$) and L-smooth functions, it is known that the convergence rate $O(1/k)$ for GD of form (1.2) is guaranteed if $\eta \leq \frac{1}{L}$ [37]. This may suggest very small step-sizes in practice, which is a condition that may be violated in more complicated scenarios. In contrast, AEGD is energy stable for any $\eta > 0$ (see Theorem 3.1), and converges for a large range of η (see Theorem 3.3).

3.1. Unconditional energy stability.

THEOREM 3.1. *(Energy stability and convergence) Consider* $\min\{f(\theta),\ \theta \in \mathbb{R}^n\}$, *where* $f(\theta)$ *is differentiable and bounded from below so that* $f(\theta) + c > 0$ *for some* $c > 0$. *Then*
(i) AEGD (2.3) is unconditionally energy stable in the sense that for any step size $\eta > 0$,

$$r_{k+1}^2 = r_k^2 - (r_{k+1} - r_k)^2 - \eta^{-1}\|\theta_{k+1} - \theta_k\|^2, \tag{3.1}$$

r_k *is strictly decreasing and convergent with* $r_k \to r^*$ *as* $k \to \infty$, *and also*

$$\sum_{j=0}^{\infty} \|\theta_{j+1} - \theta_j\|^2 \leq \eta(r_0^2 - (r^*)^2), \quad \textit{hence} \quad \lim_{k\to\infty} \|\theta_{k+1} - \theta_k\| = 0. \tag{3.2}$$

(ii) AEGD (2.4) is unconditionally energy stable in the sense that for any step size $\eta > 0$,

$$r_{k+1,i}^2 = r_{k,i}^2 - (r_{k+1,i} - r_{k,i})^2 - \eta^{-1}(\theta_{k+1,i} - \theta_{k,i})^2, \quad i \in [n], \tag{3.3}$$

$r_{k,i}$ is strictly decreasing and convergent with $r_{k,i} \to r_i^*$ as $k \to \infty$, and also

$$\sum_{j=0}^{\infty} \|\theta_{j+1} - \theta_j\|^2 \leq \eta \sum_{i=1}^{n} (r_{0,i}^2 - (r_i^*)^2), \quad \textit{hence} \quad \lim_{k\to\infty} \|\theta_{k+1} - \theta_k\| = 0. \tag{3.4}$$

Proof. (i) We use the two equations in (2.2) to derive

$$2r_{k+1}(r_{k+1} - r_k) = 2r_{k+1}\nabla g(\theta_k)\cdot(\theta_{k+1} - \theta_k) = -4\eta r_{k+1}^2 \|\nabla g(\theta_k)\|^2 = -\frac{1}{\eta}\|\theta_{k+1} - \theta_k\|^2.$$

Upon rewriting with $2b(b-a) = b^2 - a^2 + (b-a)^2$ we obtain equality (3.1). From this we see that r_k^2 is monotonically decreasing (also bounded below), therefore convergent; so does r_k since $r_k \geq 0$. Summation of (3.1) over k from $0, 1, \cdots$ yields (3.2). The proof of (ii) is entirely similar. ∎

Remark 3.2. It is worth pointing out that this theorem does not require the function f satisfy L-smoothness or convexity assumption. The result asserts that the unconditional energy stability featured by AEGD also implies convergence of $\{r_k\}_{k\geq 0}$ for any $\eta > 0$, and the sequence $\{\|\theta_{k+1} - \theta_k\|\}_{k\geq 0}$ converges to zero at a rate of at least $1/\sqrt{k}$. But this is not sufficient—at least in general—to prove convergence of the sequence $\{\theta_k\}_{k\geq 0}$, when no further information is available about this sequence.

3.2. Convergence and convergence rates. To understand the convergence behavior of AEGD (2.3), we reformulate it as

$$\theta_{k+1} = \theta_k - \eta_k \nabla f(\theta_k), \quad \eta_k := \eta \frac{r_{k+1}}{g(\theta_k)}. \tag{3.5}$$

Note that using L-smoothness of f and (3.5) we have

$$f(\theta_{k+1}) \leq f(\theta_k) - \left(\frac{1}{\eta_k} - \frac{L}{2}\right)\|\theta_{k+1} - \theta_k\|^2 < f(\theta_k),$$

when η_k gets smaller so that $\eta_k < 2/L$. Since r_k is decreasing, after finite number of iterations η_j can be ensured (by choosing η to be suitably small if necessary) to fall below $2/L$ under which $f(\theta_j)$ turns into a strictly decreasing sequence, hence convergent. To ensure the convergence of $\{\theta_k\}$, one needs further info on the geometrical property of the objective function, such as convexity, or the Kurdyka-Lojasiewicz (KL) property. The KL property at a point describes how the objective function can be made sharp through a concave mapping near that point. This property characterizes a rich function class, and often considered as a structural assumption in general nonconvex optimization (see [2] for a detailed account on this property and its applications). In the present work we restrict to a special case of KL, called Polyak-Lojasiewics (PL) property: f is PL if there exists $\mu > 0$ such that

$$\frac{1}{2}\|\nabla f(\theta)\|^2 \geq \mu(f(\theta) - f^*), \tag{3.6}$$

where we assume that $\{\theta, f(\theta) = f^*\}$ is not empty. One such example is $f(x) = x^2 + 3sin^2(x)$, non-convex, yet satisfying the PL inequality with $f^* = 0$ and $\mu = 1/32$.

In Theorem 3.3 below, we present convergence rates of AEGD in two different cases: nonconvex with the PL property and convex. For general η, convergence rates can depend on the behavior of r_k, which is part of the solution to the AEGD algorithm. Therefore, we present a hybrid result of convergence rates combining both k and r_k (posterior estimates).

THEOREM 3.3. *(Convergence rates) Suppose f is differentiable and bounded from below. Let θ_k be the k-th iterate generated by AEGD (3.5), then*

$$\frac{1}{k}\sum_{j=0}^{k-1}\|\nabla g(\theta_j)\|^2 \le \frac{\sqrt{f(\theta_0)+c}}{2\eta k r_k}.$$

We have convergence rates in two distinct cases:
(i) f is PL and L-smooth with a minimizer θ^*. If $\max_{k_0\le j<k}\eta_j \le 1/L$ for some $k_0 \ge 0$, then $\{\theta_k\}$ is convergent. Moreover,

$$\sum_{k=k_0}^{\infty}\|\theta_{k+1}-\theta_k\| \le \frac{4}{\sqrt{2\mu}}\sqrt{f(\theta_{k_0})-f(\theta^*)}, \tag{3.7}$$
$$f(\theta_k)-f(\theta^*) \le e^{-c_0(k-k_0)r_k}(f(\theta_{k_0})-f(\theta^*)), \quad c_0 := \frac{\mu\eta}{\sqrt{f(\theta_{k_0})+c}}.$$

(ii) f is convex and L-smooth with a global minimizer θ^*. If $\max_{k_0\le j<k}\eta_j \le 1/L$ for some $k_0 \ge 0$, then

$$f(\theta_k)-f(\theta^*) \le \frac{c_1\|\theta_{k_0}-\theta^*\|^2}{2(k-k_0)r_k},\ c_1 := \frac{\sqrt{f(\theta_{k_0})+c}}{\eta}.$$

See Appendix A.1 for the proof.

Remark 3.4. The gradient estimate in Theorem 3.3 when using $g(\theta)=\sqrt{f(\theta)+c}$ gives

$$\frac{1}{k}\sum_{j=0}^{k-1}\|\nabla f(\theta_j)\|^2 \le \frac{2(f(\theta_0)+c)^{\frac{1}{2}}(F_k+c)}{\eta k r_k}, \quad F_k = max_{j<k}f(\theta_j).$$

For this estimate, neither L-smoothness nor step size restriction is needed. This is in sharp contrast to the classical result for GD (see pages 29-31 in [37]).

Remark 3.5. Regarding the convergence rates, a series of remarks are in order.

1. The convergence rate in (i) may be extended to the case when f features the more general KL property; see [2, 1] for relevant techniques in establishing convergence rates for proximal and other gradient methods.
2. If $r^* > 0$, which is the case to be shown in Lemma 3.3 for $\eta < \tau$, then r_k on the right of the three estimates in Theorem 3.3 can all be replaced by $r^*(< r_k)$, hence (i) guarantees the linear convergence rate when f is PL; and (ii) recovers the usual sublinear rate $O(\frac{1}{k})$.
3. It is clear that all results are valid as long as $r_k \to 0$ slower than $1/k$. Our numerical tests indicate that for each objective function f there exists a threshold $\tilde{\eta}$ in the sense that convergence is ensured if $\eta \le \tilde{\eta}$. However, identifying $\tilde{\eta}$ appears to be a challenging task in theory.

Remark 3.6. Regarding the base step size η, we make further remarks.

1. Empirical evidence (see Figure 3 (c)) shows that there exists a threshold index J so that η_j turns to decrease for $j > J$. Hence the assumptions in (i)-(ii) are reasonable and readily met.
2. Our results suggest that decaying η at a later stage (say $k \ge k_0$ for some k_0) to ensure the sufficient conditions in Theorem 3.3 can help to achieve all-time good performance of the AEGD algorithm. This comment applies to the element-wise AEGD as well (see Theorem 3.10).

Remark 3.7. If f is μ-strongly convex, then f is also PL:

$$\begin{aligned} f(\theta^*) &\geq f(\theta) + \nabla f(\theta)\cdot(\theta^*-\theta) + \frac{\mu}{2}\|\theta^*-\theta\|^2 \\ &= f(\theta) - \frac{1}{2\mu}\|\nabla f(\theta)\|^2 + \frac{\mu}{2}\|\frac{1}{\mu}\nabla f(\theta) + (\theta^*-\theta)\|^2 \\ &\geq f(\theta) - \frac{1}{2\mu}\|\nabla f(\theta)\|^2. \end{aligned}$$

Hence result in (i) holds true for strongly convex f.

3.3. Behavior of the energy. Since r_k is strictly decreasing and $g(\theta_k)$ is positive and bounded from below and above, their relative ratio η_k essentially depends on the behavior of r_k. On the other hand, as $\eta \to 0$, the numerical solution (θ_k, r_k) may be shown to converge to the solution of the ODE system

$$\dot\theta = -2r\nabla g, \quad \dot r = \nabla g\cdot\dot\theta \tag{3.8}$$

at $t_k = k\eta$, subject to initial data $\theta(0)=\theta_0, r(0)=g(\theta_0)$. For this system the level set $r(t)-g(\theta(t))=0$ is invariant for all time. Hence for a fixed but suitably small η, starting from $(\theta_0, g(\theta_0))$, the limit of (θ_k, r_k) as $k\to\infty$ must be approaching to $(\theta^*, g(\theta^*))$, that is

$$r^* \approx g(\theta^*) > 0.$$

A natural question is whether a threshold τ for the base step size η can be identified so that we will still have $r^*>0$ for $\eta\leq\tau$. Indeed, we are able to obtain a sufficient condition to ensure this.

LEMMA 3.8. *Suppose f is L-smooth, bounded from below by $f(\theta^*)$ and we have $\max\|\nabla f(\theta)\| \leq G_\infty$, then g is L_g-smooth with*

$$g \geq g(\theta^*) = \sqrt{f(\theta^*)+c}, \quad L_g = \frac{1}{2g(\theta^*)}\left(L + \frac{G_\infty^2}{2g^2(\theta^*)}\right).$$

Consider AEGD (2.3), then

$$r_k > r^* > g(\theta^*)(1-\eta/\tau), \quad \tau := \frac{2g(\theta^*)}{L_g g^2(\theta_0)}. \tag{3.9}$$

This implies that $r^*>0$ if $\eta\leq\tau$.

Proof. For any $x, y\in\mathbb{R}^n$ we have

$$\begin{aligned} \|\nabla g(x)-\nabla g(y)\| &= \frac{1}{2}\left\|\frac{\nabla f(x)(g(y)-g(x))}{g(x)g(y)} + \frac{\nabla f(x)-\nabla f(y)}{g(y)}\right\| \\ &\leq \frac{G_\infty}{2g^2(\theta^*)}|g(y)-g(x)| + \frac{1}{2g(\theta^*)}\|\nabla f(x)-\nabla f(y)\| \leq L_g\|x-y\|. \end{aligned}$$

Such L_g-smoothness of g implies that

$$\begin{aligned} g(\theta_{j+1}) &\leq g(\theta_j) + \nabla g(\theta_j)\cdot(\theta_{j+1}-\theta_j) + \frac{L_g}{2}\|\theta_{j+1}-\theta_j\|^2 \\ &= g(\theta_j) + r_{j+1} - r_j + \frac{L_g}{2}\|\theta_{j+1}-\theta_j\|^2. \end{aligned}$$

Take a summation over j from 0 to $k-1$ so that

$$\begin{aligned} g(\theta_k) - g(\theta_0) &\le r_k - r_0 + \frac{L_g}{2}\sum_{j=0}^{k-1}\|\theta_{j+1}-\theta_j\|^2 \\ &= r_k - r_0 + \frac{L_g\eta}{2}\sum_{j=0}^{k-1}\left(r_j^2 - r_{j+1}^2 - (r_{j+1}-r_j)^2\right) \\ &= r_k - r_0 + \frac{L_g\eta}{2}\left(r_0^2 - r_k^2 - \sum_{j=0}^{k-1}(r_{j+1}-r_j)^2\right), \end{aligned}$$

where (3.1) was used. Using $r_0 = g(\theta_0)$ and $g(\theta_k) \ge g(\theta^*)$, we have for any k,

$$\frac{L_g\eta}{2}\sum_{j=0}^{k-1}(r_{j+1}-r_j)^2 + \frac{L_g\eta}{2}r_k^2 - r_k + g(\theta^*) - \frac{L_g\eta g^2(\theta_0)}{2} \le 0.$$

Passing to the limit as $k \to \infty$, we also have

$$\frac{L_g\eta}{2}\sum_{j=0}^{\infty}(r_{j+1}-r_j)^2 + \frac{L_g\eta}{2}(r^*)^2 - r^* + g(\theta^*) - \frac{\eta L_g g^2(\theta_0)}{2} \le 0.$$

Since the first two terms are positive, we have

$$r^* > g(\theta^*) - \frac{\eta L_g g^2(\theta_0)}{2} = g(\theta^*)(1-\eta/\tau),$$

where the definition of τ in (3.9) was used. ∎

Remark 3.9. 1. Note that $\eta \le \tau$ is only a sufficient condition, not necessary for $r^* > 0$ to surely happen.
2. One may take a suitably large η to gain initial rapid progress, and adjust η at a later stage at $k = k_0$. A similar argument shows that $r^* > 0$ is ensured by $\eta < \tau_1$ for $k \ge k_0$,

$$\eta < \tau_1 := \frac{2(g(\theta^*) + r_{k_0} - g(\theta_{k_0}))}{L_g r_{k_0}^2}. \tag{3.10}$$

As we expected, for small η, AEGD features same convergence rates as GD does, since the rates are essentially depending on the local geometry of f near θ^*.

3.4. Convergence results for the element-wise AEGD. Similar results also hold for the element-wise AEGD (2.4), although analysis is more involved. With the notation

$$\eta_{ij} := \eta r_{j+1,i}/g(\theta_j),\ i \in [n], \quad j = 0, 1, 2, \cdots,$$

we now present the main result for (2.4) in the following.

THEOREM 3.10. *(Convergence rates) Suppose f is differentiable and bounded from below. Let θ_k be the k-th iterate generated by the AEGD (2.4), then*

$$\min_{j<k}(\partial_i g(\theta_j))^2 \le \frac{1}{k}\sum_{j=0}^{k-1}(\partial_i g(\theta_j))^2 \le \frac{\sqrt{f(\theta_0)+c}}{2\eta k r_{k,i}}, \quad i \in [n].$$

We have convergence rates in two distinct cases:
(i) f is PL and L-smooth with a minimizer θ^*. If $\max_{k_0\le j\le k}\eta_{ij}\le\frac{1}{L}$ for $i\in[n]$ and some $k_0\ge 0$, then

$$f(\theta_k)-f(\theta^*)\le e^{-c_0(k-k_0)\min_i r_{k,i}}(f(\theta_{k_0})-f(\theta^*)),\quad c_0:=\frac{\mu\eta}{\sqrt{f(\theta_{k_0})+c}}.$$

(ii) f is convex and L-smooth with a global minimizer θ^*. If $\max_{k_0\le j<k}\eta_{ij}\le\frac{1}{L}$ for $i\in[n]$ and some $k_0\ge 0$, then

$$f(\theta_k)-f(\theta^*)\le\frac{c_1\max_{k_0\le j<k}\|\theta_j-\theta^*\|^2}{(k-k_0)\min_i r_{k,i}},\ c_1:=\frac{\sqrt{f(\theta_{k_0})+c}}{\eta}.$$

See Appendix A.3 for the proof.

Remark 3.11. The above gradient estimate when using $g(\theta)=\sqrt{f(\theta)+c}$ leads to

$$\min_{j<k}(\partial_i f(\theta_j))^2\le\frac{1}{k}\sum_{j=0}^{k-1}(\partial_i f(\theta_j))^2\le\frac{2(f(\theta_0)+c)^{\frac{1}{2}}(F_k+c)}{\eta k r_{k,i}},\quad F_k=max_{j<k}f(\theta_j).$$

As argued above, our convergence rates in Theorem 3.10 depend also on the behavior of $r_{k,i}$, about which we have the following result.

LEMMA 3.12. *Under the same assumptions as in Lemma 3.3, for AEGD (2.4) we have for $i\in[n]$,*

$$r_{k,i}>r_i^*>g(\theta^*)(1-\eta/\tilde\tau),\quad \tilde\tau:=\frac{2g(\theta^*)}{nL_g(g(\theta_0))^2}.$$

This implies that $r_i^*>0$ if $\eta\le\tilde\tau$.

We include a proof in Appendix A.2.

4. Stochastic AEGD. This section presents a stochastic AEGD for the unconstrained finite-sum optimization problem:

$$\min_\theta\left\{f(\theta)=\frac{1}{m}\sum_{i=1}^m f_i(\theta)\right\}. \tag{4.1}$$

We use θ^* to denote a minimizer of $f(\theta)$, and assume that f_i is differentiable and lower bounded so that $f_i(\theta)>-c$, for $i\in[m]$, for some $c>0$. This problem is prevalent in machine learning tasks where θ corresponds to the model parameters, $f_i(\theta)$ represents a loss on the training point i and the aim is to minimize the average loss $f(\theta)$ across points. When m is large, SGD or its variants are preferred for solving (4.1) mainly because of their cheap per iteration cost. To present a stochastic algorithm, we use v_k to denote a random search direction at k-th step. Here we give only the element-wise version of our stochastic AEGD.

Remark 4.1. If at each iteration step, a mini-batch of training data were selected, Algorithm 4.1 still applies if $f_{i_k}(\theta_k)$ is replaced by $\frac{1}{b}\sum_{i\in B_k}f_i(\theta_k)$, where B_k denotes a randomly selected subset of $[m]$ of size b at step k with $b\ll m$.

To allow for any form of minibatching we use the arbitrary sampling notation

$$f_\xi(\theta)=\frac{1}{m}\sum_{j=1}^m\xi_j f_j(\theta),$$

Algorithm 4.1 Stochastic AEGD. Good default setting for parameters are $c = 1$ and $\eta = 0.1$.

Require: $\{f_i(\theta)\}_{i=1}^m$, η: the step size, θ_0: initial guess of θ, and K: the total number of iterations.

Require: c: a parameter such that for any $i \in [m]$, $f_i(\theta) + c > 0$ for all $\theta \in \mathbb{R}^n$, initial energy: $r_0 = \sqrt{f_{i_0}(\theta_0) + c}\mathbf{1}$

1: **for** $k = 0$ to $K - 1$ **do**
2: $v_k := \nabla f_{i_k}(\theta_k)/\big(2\sqrt{f_{i_k}(\theta_k) + c}\big)$ (i_k is a random sample from $[m]$ at step k)
3: $r_{k+1} = r_k/(1 + 2\eta v_k \odot v_k)$ (update energy)
4: $\theta_{k+1} = \theta_k - 2\eta r_{k+1} \odot v_k$
5: **end for**
6: **return** θ_K

where $\xi = (\xi^1, \cdots, \xi^m) \in \mathbb{R}^m_+$ is a random sampling vector (drawn from some distribution) such that $\mathbb{E}[\xi^j] = 1$ for $j \in [m]$. The element-wise update rule for the stochastic AEGD with arbitrary sampling can be reformulated as

$$v_k = \nabla f_{\xi_k}(\theta_k)/(2\sqrt{f_{\xi_k}(\theta_k) + c}), \tag{4.2a}$$

$$r_{k+1,i} - r_{k,i} = v_{k,i}(\theta_{k+1,i} - \theta_{k,i}), \quad r_{0,i} = \sqrt{f_{\xi_0}(\theta_0) + c} \tag{4.2b}$$

$$\theta_{k+1,i} = \theta_{k,i} - 2\eta r_{k+1,i} v_{k,i} \quad k = 0, 1, 2, \cdots. \tag{4.2c}$$

It follows immediately from the definition of sampling vector ξ that

$$\mathbb{E}[f_\xi(\theta)] = f(\theta), \quad \mathbb{E}[\nabla f_\xi(\theta)] = \nabla f(\theta),$$

which means that we still have access to unbiased estimates of f and its gradient. Of particular interest is the minibatch sampling: $\xi \in \mathbb{R}^m_+$ is a b-minibatch sampling if for every subset $B \in [m]$ with $|B| = b$ we have that

$$\mathbb{P}\left[\xi = \frac{m}{b}\sum_{i\in B} e_i\right] = \frac{b!(m-b)!}{m!}.$$

One can show by a double counting argument that if ξ is a $b-$minibatch sampling, it is indeed a valid sampling with $\mathbb{E}[\xi^j] = 1$ (see [19]) and $\frac{1}{m}\sum_{j=1}^m \xi^j = 1$.

No matter how v_k is defined, we have the following result.

THEOREM 4.2. *(Unconditional energy stability) The stochastic AEGD of form (4.2) is unconditionally energy stable in the sense that for any step size $\eta > 0$,*

$$\mathbb{E}[r^2_{k+1,i}] = \mathbb{E}[r^2_{k,i}] - \mathbb{E}[(r_{k+1,i} - r_{k,i})^2] - \eta^{-1}\mathbb{E}[(\theta_{k+1,i} - \theta_{k,i})^2], \quad i \in [n], \tag{4.3}$$

that is $\mathbb{E}[r_{k,i}]$ is strictly decreasing and convergent with $\mathbb{E}[r_{k,i}] \to r_i^*$ as $k \to \infty$, and also

$$\sum_{j=0}^{\infty} \mathbb{E}[|\theta_{j+1,i} - \theta_{j,i}|^2] \le \eta(f(\theta_0) + c), \text{ hence } \lim_{k\to\infty} \mathbb{E}[|\theta_{k+1,i} - \theta_{k,i}|^2] = 0, \quad \forall i \in [n]. \tag{4.4}$$

The proof is entirely similar to that for Theorem 3.1, details are omitted.

With mild assumptions on f_l we are able to establish the following.

THEOREM 4.3. *Suppose* $\|\nabla f_l\|_\infty \le G_\infty$, *and* $f_l + c \ge a > 0$ *for all* $l \in [m]$. *Then stochastic AEGD of form (4.2) admits the following direction-wise estimate, for* $i \in [n]$,

$$\frac{1}{k}\sum_{j=0}^{k}\mathbb{E}[(v_{j,i})^2] \le \frac{C_i}{k\mathbb{E}[r_{k,i}]}, \quad C_i := \frac{\mathbb{E}[r_{0,i}](2a+\eta G_\infty^2)}{4a\eta}.$$

The proof is given in Appendix A.4.

As in the deterministic case, the above rough bound comes directly from the special structure of the scheme. Due to the nonlinearity of v_k in terms of the random variables, we only have $2\mathbb{E}\left[v_k\sqrt{f_{\xi_k}(\theta_k)+c}\right] = \nabla f(\theta_k)$, the estimate of refined convergence rates is more subtle, and will be dealt with elsewhere [30].

5. Experimental Results. We evaluate the deterministic AEGD (2.4) and stochastic AEGD (Algorithm 4.1) on several benchmarks for optimization, including convex and non-convex performance testing problems, k-means clustering, which has a non-smooth objective function, and convolutional neural networks on the standard CIFAR-10 and CIFAR-100 data sets. Overall, we show that AEGD is a versatile algorithm that can efficiently solve a variety of optimization problems.

In all experiments, we fine tune the base step size for each algorithm to obtain its best performance. The base step size is given in respective plots, denoted as "lr", and the momentum in GDM and stochastic GDM (SGDM) is set to 0.9. For AEGD we found that the parameters that impact performance the most were the base step size and step decay schedule in the deep learning experiments. Parameter c has little impact on the method performance (see a test in Section 5.1), we simply take c as a fixed value as long as $f + c > 0$. We use $c = 1$ for nonnegative objective functions.

Our experiments show the following primary findings: (i) AEGD allows larger effective step size, hence evolves much faster than GD; (ii) Empirically the performance of AEGD appears better than or at least comparable with (S)GDM: in full batch setting, AEGD typically displays rapid initial progress while GDM tends to overshoot and detour to the target; in stochastic settings, AEGD produces solutions that generalize better than SGDM when coupled with a learning rate decay schedule.

5.1. Performance Testing Problems. We begin with two benchmark convex and non-convex performance testing problems:

(i) A quadratic function (a strongly convex function with $\mu = 2/100$ and $L = 2$)

$$f(x_1, x_2, ..., x_{100}) = \sum_{i=1}^{50} x_{2i-1}^2 + \sum_{i=1}^{50}\frac{x_{2i}^2}{10^2}, \tag{5.1}$$

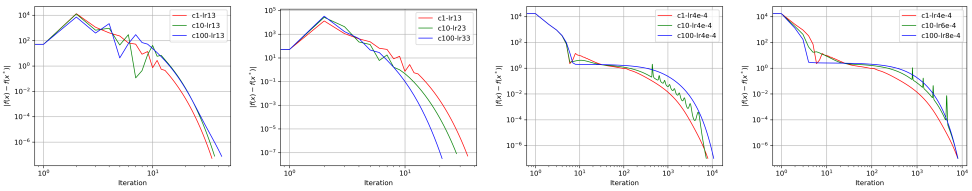
with initial point $(1, 1, ..., 1)$;

(ii) The 2D Rosenbrock function (non-convex and L-smooth)

$$f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2, \tag{5.2}$$

with initial point $(-3, -4)$.

Effect of c: Using these examples we first show the marginal effect of c on the performance of AEGD. For AEGD with $c = 1, 10, 100$, we search and take the base step size η that gives the best performance (fastest convergence) in each case. The results presented in Figure 1 (b) (d) indicate that a larger c would require a larger base step size to achieve faster initial progress, but which c to use does not seem to affect the overall performance of AEGD significantly. We also present the results of



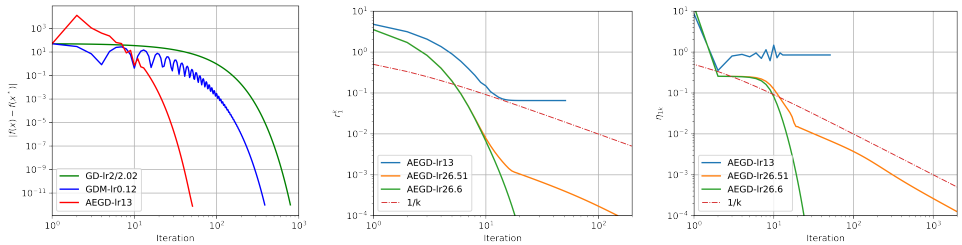
(a) (b) (c) (d)

Fig. 1: The effect of c on the performance of AEGD when applied to the quadratic problem (a)(b) and the Rosenbrock problem (c)(d).

AEGD with different c but the same base step size in 1 (a) (c), from which we see that the differences between the results are marginal.

Now we compare the performance of AEGD with GD and GDM when applied to the two benchmark problems. Numerical comparison results in Figure 2 (a) are for the quadratic function. We see that AEGD converges much faster than both GD and GDM on this problem.

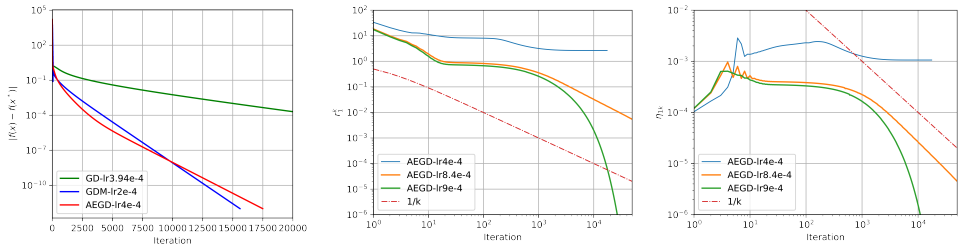
Results in Figure 3 (a) are for the Rosenbrock function. For this problem, the convergence of GD is very slow compared with GDM and AEGD. Though the number of iterations that AEGD needs to reach the minima is slightly more than GDM, AEGD makes a faster initial progress. This can be observed more clearly in Figure 4, which shows that GD and GDM tend to overshoot and detour to the minima while AEGD goes along a more direct path to the minima.



(a) (b) (c)

Fig. 2: The quadratic problem: (a) Optimality gap of different algorithms; (b) Behavior of $r_{k,1}$ for different η; (c) Behavior of η_{1k} for different η.

Behavior of r_k: We also numerically investigated how the base step size η affects the behavior of r_k of AEGD on the two problems. (For both functions, $r_{k,1} = \min_i r_{k,i}$, hence we only show the behavior of $r_{k,1}$.) The results are presented in Figure 2 (b) for the quadratic problem and Figure 3 (b) for the Rosenbrock problem. From these results, there appears to exist a threshold $\tilde{\eta}$ such that when $\eta < \tilde{\eta}$, $r^* > 0$; when $\eta > \tilde{\eta}$, $r^* = 0$; when $\eta = \tilde{\eta}$, r_k converges to 0 at the speed rate of $O(1/k)$; and when $\eta \leq \tilde{\eta}$, AEGD converges to the minima. We conjecture that such critical threshold phenomenon for the base step size η should hold true for more general objective



(a) (b) (c)

Fig. 3: The Rosenbrock problem: (a) Optimality gap of different algorithms; (b) Behavior of $r_{k,1}$ for different η; (c) Behavior of η_{1k} for different η.

functions. Specifically, for the quadratic problem: $\tilde{\eta} \sim 26.51$; for the Rosenbrock problem, $\tilde{\eta} \sim 8.4e-4$. Based on our tests, the largest step size GD ensures convergence is 1 for the quadratic problem and $\sim 3.94e-4$ for the Rosenbrock problem. In both cases, $\tilde{\eta}$ is much larger than the admissible step sizes for GD.

Behavior of η_k: From Figure 2 (c) and Figure 3 (c), we see that η_k can increase in the initial stage, and turn to decrease at a later stage – showing a threshold phenomenon as argued in Remark 3.6. At the final stage $\eta_k \sim \frac{\eta}{\sqrt{c+f^*}} r_{k+1}$, i.e, the convergence behavior of η_k is dominated by the convergence behavior of r_k, regardless of the size of η.

We note that GDM has been proven to be faster than GD in terms of convergence rates in certain cases; see [44] for the quadratic convex problem, While AEGD shares the same convergence rates as GD. However, both GD and GDM suffer from step-size limitations. In contrast, AEGD allows larger effective step sizes η_k in the initial stage, which can result in faster initial progress than GDM. This is more observable for problems with large condition numbers, as shown in Figure 2 (a) and Figure 3 (a).

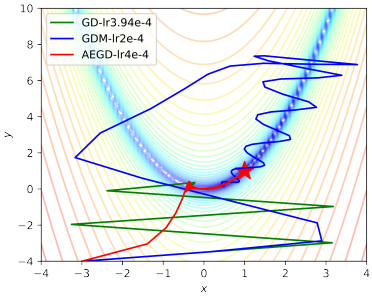


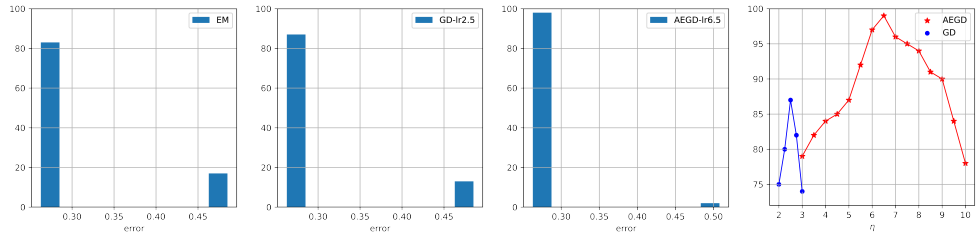
Fig. 4: Trajectories of different optimization algorithms on the Rosenbrock function.

5.2. K-means Clustering. We consider the k-means clustering problem for a set of data points $\{\boldsymbol{p}_i\}_{i=1}^m$ in $\mathbb{R}^d$ with K centroids $\{\boldsymbol{x}_j\}_{j=1}^K$. Denote $\boldsymbol{x} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K] \in$

$\mathbb{R}^{Kd}$, we seek to minimize the quantization error:

$$\min_{\boldsymbol{x}\in\mathbb{R}^{Kd}} \left\{ f(\boldsymbol{x}) := \frac{1}{2m} \sum_{i=1}^{m} \min_{1\le j\le K} \|\boldsymbol{x}_j - \boldsymbol{p}_i\|^2 \right\}. \tag{5.3}$$

If a data point $\boldsymbol{p}_i$ has more than one distinct nearest centroids, we assign $\boldsymbol{p}_i$ to one of them randomly. We define the gradient of f (presented in Appendix B.1) as [10] did when applying the gradient-based method. In this example, the Iris data set, which contains 150 four-dimensional data samples from 3 categories, is used to compare the robustness to initialization of three methods: GD, AEGD, and expectation maximization (EM) [33].



(a) (b) (c) (d)

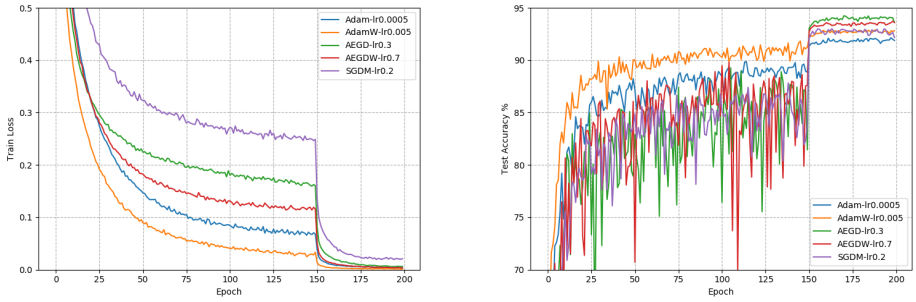
Fig. 5: The histogram of the quantization error of k-means on Iris trained by EM (a), GD (b) and AEGD (c) over 100 independent experiments. (d) Frequency of GD and AEGD achieve the improved minimum valued at ~ 0.26 in 100 runs with different base step sizes.

Figure 5 (a) (b) (c) present the frequency of error given by the three methods in 100 runs. In each run, the initial centroids are selected from the data set randomly. We see that though there are chances for all the three methods to get stuck at a local minimum whose value is ~ 0.48, AEGD managed to locate an improved minimum valued at ~ 0.26 with the highest probability.

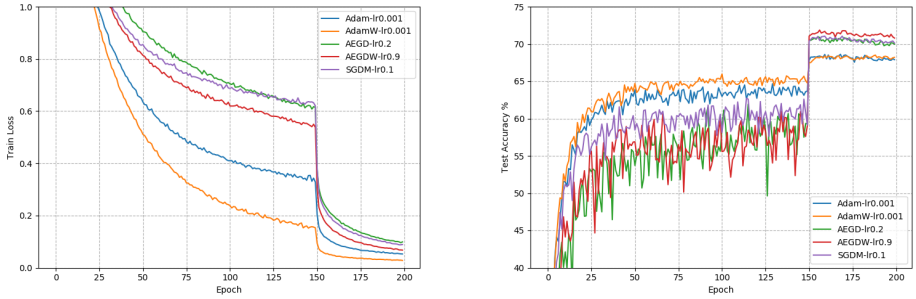
We also present the frequency of GD and AEGD achieving the improved minimum in 100 runs with different base step sizes in Figure 5 (d). We see that compared with GD, AEGD allows a larger set of base step size to achieve the improved minimum with much higher probability, with $\eta \sim 6.5$ being the optimal choice.

5.3. Convolutional Neural Networks. First, we should point out that the generalization capability of AEGD in training deep neural networks remains to be further understood. However, we would like to present some preliminary results to show the potential of the AEGD in this aspect.

Using ResNet-56 [20] and SqueezeNet [21], we consider the task of image classification on the standard CIFAR-10 and CIFAR-100 datasets. In our experiments, we employ the fixed budget of 200 epochs and reduce the learning rates by 10 after 150 epochs, with a minibatch size of 128 and weight decay of 1×10^{-4}. We recall that weight decay has been a standard trick in training neural networks [8, 26]. For SGD, it can be interpreted as a form of L^2 regularization, but for adaptive algorithms such as Adam, careful implementation techniques are often needed [35, 59]. Inspired by



(a) Training loss, ResNet-56, CIFAR-10 (b) Test accuracy, ResNet-56, CIFAR-10



(c) Training loss, SqueezeNet, CIFAR-100 (d) Test accuracy, SqueezeNet, CIFAR-100

Fig. 6: Training loss and test accuracy for ResNet-56 on CIFAR-10 and SqueezeNet on CIFAR-100

[35], we introduce an AEGD-specific weight decay algorithm, called AEGDW. The algorithm for AEGDW and the initial set of step sizes are presented in Appendix B.2.

Our experimental results, as given in Figure 6, show that Adam and AdamW perform better than other algorithms early. But by epoch 150 when the learning rates are decayed, AEGD and AEGDW significantly outperform other methods in generalization. In the above two experiments, AEGD(W) even surpasses SGDM by 1% in test accuracy.

6. Conclusions. Inspired by the IEQ approach for gradient flows in the form of time-dependent partial differential equations, we proposed AEGD in both global and element-wise form, as a new algorithm for optimization of non-convex objective functions. This simple update with an auxiliary energy variable is easy to implement, proven to be unconditionally energy stable irrespective of the base step size, and features energy-dependent convergence rates with mild conditions on the base step size. It is suitable for general objective functions, as long as they are bounded from below. Numerical examples ranging from performance test problems, K-means clustering, to CNNs, all demonstrate the advantages of the proposed AEGD: it enjoys rapid initial progress and faster convergence than GD, is robust with respect to initial data, and generalizes better for deep learning problems. Overall, the results presented in this

paper suggest that further research of AEGD could prove useful.

Appendix A. Technical Proofs. In this section, we prove Theorem 3.3 and Theorem 3.10 from Section 3.

A.1. Proof of Theorem 3.3. We rewrite AEGD (2.3) as

$$\theta_{k+1} = \theta_k - \eta_k \nabla f(\theta_k), \quad \eta_k := \eta \frac{r_{k+1}}{g(\theta_k)}. \tag{A.1}$$

This is in the form of the usual GD with a variable step size η_k.
(i) Using scheme (2.2) we have

$$r_{j+1} - r_j = \nabla g(\theta_j) \cdot (\theta_{j+1} - \theta_j) = -2\eta r_{j+1} \|\nabla g(\theta_j)\|^2.$$

Take a summation over j from 0 to $k-1$ gives

$$r_0 - r_k = 2\eta \sum_{j=0}^{k-1} r_{j+1} \|\nabla g(\theta_j)\|^2 \geq 2\eta r_k \sum_{j=0}^{k-1} \|\nabla g(\theta_j)\|^2$$

Use $r_0 = g(\theta_0)$ and r_k is strictly decreasing to get

$$k \min_{j<k} \|\nabla g(\theta_j)\|^2 \leq \sum_{j=0}^{k-1} \|\nabla g(\theta_j)\|^2 \leq \frac{g(\theta_0)}{2\eta r_k}.$$

Next we prove (i) and (ii), taking $k_0 = 0$, for simplicity in presentation.
(i) By L-smoothness of f and scheme (A.1), we have

$$\begin{aligned} f(\theta_{k+1}) - f(\theta_k) &\leq \nabla f(\theta_k) \cdot (\theta_{k+1} - \theta_k) + \frac{L}{2} \|\theta_{k+1} - \theta_k\|^2 \\ &\leq -\frac{\eta_k}{2}(2 - L\eta_k) \|\nabla f(\theta_k)\|^2 \leq -\frac{\eta_k}{2} \|\nabla f(\theta_k)\|^2 \end{aligned} \tag{A.2}$$

as long as $\eta_k \leq 1/L$. Here and in what follows, we denote $w_k = f(\theta_k) - f^*$, then the PL property reads

$$\frac{1}{2} \|\nabla f(\theta_k)\|^2 \geq \mu (f(\theta_k) - f^*) = \mu w_k.$$

With this property, (A.2) can be written as

$$w_{k+1} - w_k \leq -\mu \eta_k w_k.$$

This implies $w_{k+1} \leq (1 - \mu\eta_k) w_k$. By induction,

$$w_k \leq \prod_{j=0}^{k-1} (1 - \mu\eta_j) w_0 = \exp\left(\sum_{j=0}^{k-1} \log(1 - \mu\eta_j) \right) w_0 \leq \exp\left(-\mu \sum_{j=0}^{k-1} \eta_j \right) w_0.$$

Noticing that $\eta_j = \eta \frac{r_{j+1}}{g(\theta_j)} \geq \eta \frac{r_k}{g(\theta_0)}$, we further get

$$w_k \leq \exp(-c_0 k r_k) w_0, \quad c_0 = \frac{\mu\eta}{g(\theta_0)}.$$

To obtain the convergence of θ_k, we use scheme (A.1) to rewrite (A.2) as

$$f(\theta_k) - f(\theta_{k+1}) \geq \frac{1}{2\eta_k} \|\theta_{k+1} - \theta_k\|^2 \quad \Rightarrow \quad w_k - w_{k+1} \geq \frac{1}{2\eta_k} \|\theta_{k+1} - \theta_k\|^2.$$

The PL property when combined with (A.1) gives

$$\|\theta_{k+1}-\theta_k\|^2 \geq 2\mu\eta_k^2 w_k \quad \Rightarrow \quad \frac{1}{\sqrt{w_k}} \geq \frac{\sqrt{2\mu}\eta_k}{\|\theta_{k+1}-\theta_k\|}.$$

Using the above two inequalities and noting that $w_k \geq w_{k+1}$, we have

$$\sqrt{w_k}-\sqrt{w_{k+1}} \geq \frac{1}{2\sqrt{w_k}}(w_k-w_{k+1}) \geq \frac{\sqrt{2\mu}}{4}\|\theta_{k+1}-\theta_k\|.$$

Taking a summation over k from 0 to ∞ gives

$$\sum_{k=0}^{\infty}\|\theta_{k+1}-\theta_k\| \leq \frac{4}{\sqrt{2\mu}}\sqrt{w_0}.$$

This yields (3.7), which ensures the convergence of $\{\theta_k\}$.
(ii) By the L-smoothness assumption, scheme (A.1) and $\eta_k \leq 1/L$, as in (i) we obtain

$$f(\theta_{k+1}) \leq f(\theta_k) - \frac{1}{2\eta_k}\|\theta_{k+1}-\theta_k\|^2.$$

Denoting $d_k := \|\theta_k-\theta^*\|$, we proceed to obtain the convergence rate. By convexity of f,

$$f(\theta_k) \leq f(\theta^*) + \nabla f(\theta_k)\cdot(\theta_k-\theta^*).$$

These when combined lead to

$$\begin{aligned} w_{k+1} &\leq \frac{1}{2\eta_k}(2\eta_k\nabla f(\theta_k)\cdot(\theta_k-\theta^*) - \|\theta_{k+1}-\theta_k\|^2) \\ &= \frac{1}{2\eta_k}(-2(\theta_{k+1}-\theta_k)\cdot(\theta_k-\theta^*) - \|\theta_{k+1}-\theta_k\|^2) \\ &= \frac{1}{2\eta_k}(d_k^2-d_{k+1}^2). \end{aligned}$$

Upon summation over iteration steps, we have

$$d_0^2-d_k^2 = 2\sum_{j=0}^{k-1}\eta_i w_{i+1} \geq w_k\sum_{j=0}^{k-1}\eta_i \geq 2kw_k\frac{\eta r_k}{g(\theta_0)}.$$

Hence for $\max_{j<k}\eta_j \leq 1/L$, we have

$$f(\theta_k)-f(\theta^*) = w_k \leq \frac{\|\theta_0-\theta^*\|^2 g(\theta_0)}{2k\eta r_k}.$$

This completes the proof.

A.2. Proof of Lemma 3.12. The L_g-smoothness of g implies that

$$\begin{aligned} g(\theta_{j+1}) &\leq g(\theta_j) + \nabla g(\theta_j)\cdot(\theta_{j+1}-\theta_j) + \frac{L_g}{2}\|\theta_{j+1}-\theta_j\|^2 \\ &= g(\theta_j) + \sum_{i=1}^{n}\partial_i g(\theta_j)(\theta_{j+1,i}-\theta_{j,i}) + \frac{L_g}{2}\sum_{i=1}^{n}(\theta_{j+1,i}-\theta_{j,i})^2 \\ &\leq g(\theta_j) + \sum_{i=1}^{n}(r_{j+1,i}-r_{j,i}) + \frac{\eta L_g}{2}\sum_{i=1}^{n}(r_{j,i}^2-r_{j+1,i}^2). \end{aligned}$$

Take summation over j from 0 to $k-1$ and use $r_{0,i} = g(\theta_0)$, so that

$$g(\theta_k) - g(\theta_0) \le \sum_{i=1}^{n} r_{k,i} - ng(\theta_0) + \frac{\eta L_g}{2}\left(n(g(\theta_0))^2 - \sum_{i=1}^{n} r_{k,i}^2\right).$$

Using $g(\theta_k) \ge g(\theta^*)$, we have for any k,

$$-\sum_{i=1}^{n} r_{k,i} + g(\theta^*) + (n-1)g(\theta_0) - \frac{n\eta L_g}{2}(g(\theta_0))^2 \le 0.$$

Passing to the limit as $k \to \infty$, we also have

$$-\sum_{i=1}^{n} r_i^* + g(\theta^*) + (n-1)g(\theta_0) - \frac{n\eta L_g}{2}(g(\theta_0))^2 \le 0.$$

From this we get

$$-\left(\min_i r_i^* + (n-1)g(\theta_0)\right) + g(\theta^*) + (n-1)g(\theta_0) - \frac{n\eta L_g}{2}(g(\theta_0))^2 \le 0,$$

which can be reduced to

$$-\min_i r_i^* + g(\theta^*)(1 - \eta/\tilde{\tau}) \le 0, \quad \tilde{\tau} := \frac{2g(\theta^*)}{nL_g(g(\theta_0))^2}.$$

Hence

$$\min_i r_i^* > g(\theta^*)(1 - \eta/\tilde{\tau}).$$

A.3. Proof of Theorem 3.10 . We rewrite AEGD (2.4) for $i \in [n]$ as

$$\theta_{k+1,i} = \theta_{k,i} - \eta_{ik}\partial_i f(\theta_k), \quad \eta_{ik} := \eta\frac{r_{k+1,i}}{g(\theta_k)}. \tag{A.3}$$

(i) Using scheme (2.5), for $i \in [n]$ we have

$$r_{j+1,i} - r_{j,i} = \partial_i g(\theta_j)(\theta_{j+1,i} - \theta_{j,i}) = -2\eta r_{j+1,i}(\partial_i g(\theta_j))^2.$$

Take summation over j from 0 to $k-1$ gives

$$r_{0,i} - r_{k,i} = 2\eta\sum_{j=0}^{k-1} r_{j+1,i}(\partial_i g(\theta_j))^2 \ge 2\eta r_{k,i}\sum_{j=0}^{k-1}(\partial_i g(\theta_j))^2$$

Using $r_{0,i} = g(\theta_0)$ and $r_{j,i}$ is strictly decreasing, we get

$$k\min_{j<k}(\partial_i g(\theta_j))^2 \le \sum_{j=0}^{k-1}(\partial_i g(\theta_j))^2 \le \frac{g(\theta_0)}{2\eta r_{k,i}}.$$

Next we turn to prove (i) and (ii). For simplicity in presentation, we take $k_0 = 0$.
(i) By L-smoothness of f and scheme (A.1), we have

$$\begin{aligned} f(\theta_{k+1}) - f(\theta_k) &\le \nabla f(\theta_k)\cdot(\theta_{k+1} - \theta_k) + \frac{L}{2}\|\theta_{k+1} - \theta_k\|^2 \\ &\le -\sum_{i=1}^{n}\frac{\eta_{ik}}{2}(\partial_i f(\theta_k))^2 \le -\frac{\min_i \eta_{ik}}{2}\|\nabla f(\theta_k)\|^2 \end{aligned}$$

as long as $\eta_{ik} \leq 1/L$. As in the proof for Theorem 3.3 (i), we have

$$w_k \leq \exp\left(-\mu \sum_{j=0}^{k-1} \min_i \eta_{ij}\right) w_0.$$

Noticing that $\min_i \eta_{ij} = \min_i \eta \frac{r_{j+1,i}}{g(\theta_j)} \geq \eta \frac{\min_i r_{k,i}}{g(\theta_0)}$, we further get

$$w_k \leq \exp(-c_0 k \min_i r_{k,i}) w_0, \quad c_0 = \frac{\mu\eta}{g(\theta_0)}.$$

(ii) Using the L-smoothness assumption, scheme (2.4) and $\eta_{ik} \leq 1/L$, as in (i) we obtain

$$f(\theta_{k+1}) \leq f(\theta_k) - \sum_{i=1}^{n} \frac{1}{2\eta_{ik}} (\theta_{k+1,i} - \theta_{k,i})^2.$$

Denote $d_{ik} := \theta_{k,i} - \theta_i^*$, then by convexity of f,

$$f(\theta_k) \leq f(\theta^*) + \nabla f(\theta_k) \cdot (\theta_k - \theta^*).$$

These when combined lead to

$$\begin{aligned} w_{k+1} &\leq \sum_{i=1}^{n} \frac{1}{2\eta_{ik}} (2\eta_{ik} \partial_i f(\theta_k)(\theta_{k,i} - \theta_i^*) - (\theta_{k+1,i} - \theta_{k,i})^2) \\ &= \sum_{i=1}^{n} \frac{1}{2\eta_{ik}} (-2(\theta_{k+1,i} - \theta_{k,i})(\theta_{k,i} - \theta_i^*) - (\theta_{k+1,i} - \theta_{k,i})^2) \\ &= \sum_{i=1}^{n} \frac{1}{2\eta_{ik}} (d_{ik}^2 - d_{i,k+1}^2). \end{aligned}$$

That is

$$\frac{2\eta}{g(\theta_k)} w_{k+1} \leq \sum_{i=1}^{n} \frac{1}{r_{k+1,i}} (d_{ik}^2 - d_{i,k+1}^2).$$

This upon summation over iteration steps gives

$$\begin{aligned} \sum_{j=0}^{k-1} \frac{2\eta}{g(\theta_j)} w_{j+1} &\leq \sum_{j=0}^{k-1} \sum_{i=1}^{n} \frac{1}{r_{j+1,i}} (d_{ij}^2 - d_{i,j+1}^2) \\ &\leq \sum_{i=1}^{n} \left[\frac{d_{i0}^2}{r_{1,i}} - \frac{d_{ik}^2}{r_{k,i}} + \sum_{j=0}^{k-1} \left(\frac{1}{r_{j+1,i}} - \frac{1}{r_{j,i}} \right) d_{ij}^2 \right] =: RHS. \end{aligned}$$

Since $f(\theta_j)$ is decreasing in j, so is $g(\theta_j)$. Hence

$$\sum_{j=0}^{k-1} \frac{2\eta}{g(\theta_j)} w_{j+1} \geq 2k\eta \frac{w_k}{g(\theta_0)}.$$

On the other hand, using $r_{j,i} > r_{j+1,i}$, we have

$$\begin{aligned} RHS &\le \sum_{i=1}^{n}\left[\frac{d_{i0}^2}{r_{1,i}} - \frac{d_{ik}^2}{r_{k,i}} + \max_{j<k} d_{ij}^2 \sum_{j=0}^{k-1}\left(\frac{1}{r_{j+1,i}} - \frac{1}{r_{j,i}}\right)\right] \\ &\le \sum_{i=1}^{n}\left[\frac{d_{i0}^2}{r_{1,i}} - \frac{d_{ik}^2}{r_{k,i}} + \max_{j<k} d_{ij}^2 \left(\frac{1}{r_{k,i}} - \frac{1}{r_{0,i}}\right)\right] \\ &\le 2\sum_{i=1}^{n}\frac{\max_{j<k} d_{ij}^2}{r_{k,i}}. \end{aligned}$$

Hence

$$w_k \le \frac{g(\theta_0)}{k\eta}\sum_{i=1}^{n}\frac{\max_{j<k}|\theta_{j,i}-\theta_i^*|^2}{r_{k,i}} \le \frac{g(\theta_0)}{k\eta}\frac{\max_{j<k}\|\theta_j-\theta^*\|^2}{\min_i r_{k,i}}.$$

A.4. Proof of Theorem 4.3. From $v_{j,i} = \partial_i f_\xi(\theta_j)/(2\sqrt{f_\xi(\theta_j)+c})$, it follows $(v_{j,i})^2 \le \frac{G_\infty^2}{4a}$. Using (4.2a, b), we obtain

$$r_{j,i} - r_{j+1,i} = -v_{j,i}(\theta_{j+1,i}-\theta_{j,i}) = 2\eta r_{j+1,i}(v_{j,i})^2 = 2\eta r_{j,i}\frac{(v_{j,i})^2}{1+2\eta(v_{j,i})^2}, \tag{A.4}$$

where we used $r_{j+1,i} = \frac{r_{j,i}}{1+2\eta(v_{j,i})^2}$. Taking expectation conditioned on (θ_j, r_j) in (A.4), we have

$$r_{j,i} - \mathbb{E}[r_{j+1,i}] = 2\eta r_{j,i}\mathbb{E}\left[\frac{(v_{j,i})^2}{1+2\eta(v_{j,i})^2}\right] \ge \frac{2\eta r_{j,i}}{1+2\eta G_\infty^2/(4a)}\mathbb{E}[(v_{j,i})^2]. \tag{A.5}$$

Rearranging and taking expectations to get

$$\mathbb{E}[r_{j,i}] - \mathbb{E}[r_{j+1,i}] \ge \frac{4a\eta}{2a+\eta G_\infty^2}\mathbb{E}[r_{j,i}]\mathbb{E}[(v_{j,i})^2].$$

Summing over j from 0 to $k-1$ and using telescopic cancellation gives

$$\mathbb{E}[r_{0,i}] - \mathbb{E}[r_{k,i}] \ge \frac{4a\eta}{2a+\eta G_\infty^2}\sum_{j=0}^{k-1}\mathbb{E}[r_{j,i}]\mathbb{E}[(v_{j,i})^2] \ge \frac{4a\eta}{2a+\eta G_\infty^2}\mathbb{E}[r_{k,i}]\sum_{j=0}^{k-1}\mathbb{E}[(v_{j,i})^2].$$

That is

$$k\min_{j<k}\mathbb{E}[(v_{j,i})^2] \le \sum_{j=0}^{k-1}\mathbb{E}[(v_{j,i})^2] \le \frac{C_i}{\mathbb{E}[r_{k,i}]}, \quad C_i := \frac{\mathbb{E}[r_{0,i}](2a+\eta G_\infty^2)}{4a\eta}.$$

This completes the proof.

Appendix B. Additional Experimental Results and Implementation Details. Here we provide additional experimental results and implementation details beyond those in Section 5.

B.1. K-means Clustering. By abuse of notation, we can define the 'gradient' of f at any point $\boldsymbol{x}$ as

$$\nabla f(\boldsymbol{x}) = \frac{1}{m}\left[\sum_{i\in\mathcal{C}_1}(\boldsymbol{x}_1-\boldsymbol{p}_i), ..., \sum_{i\in\mathcal{C}_K}(\boldsymbol{x}_K-\boldsymbol{p}_i)\right]^\top, \tag{B.1}$$

where $\mathcal{C}_j$ denotes the index set of the points that are assigned to the centroid $\boldsymbol{x}_j$. With the definition of loss function in (5.3) and its gradient so defined, we can apply gradient-based methods including AEGD to solve the $k-$means clustering problem.

Algorithm B.1 AEGD with decoupled weight decay (AEGDW). Good default setting for parameters are $c = 1$ and $\eta = 0.7/0.9$ for deep learning problems

Require: $\{f_j(\theta)\}_{j=1}^m$, η: the step size, θ_0: initial guess of θ, and K: the total number of iterations.

Require: c: a parameter such that $f(\theta) + c > 0$ for all $\theta \in \mathbb{R}^n$, initial energy, $r_0 = \sqrt{f(\theta_0) + c}\mathbf{1}$, weight decay factor $\lambda \in \mathbb{R}$

for $k = 0$ to $K - 1$ **do**

  $v_k := \nabla f_{i_k}(\theta_k)/\big(2\sqrt{f_{i_k}(\theta_k) + c}\big)$ (i_k is a random sample from $[m]$ at step k)

  $r_{k+1} = r_k/(1 + 2\eta v_k \odot v_k)$ (update energy)

  $\theta_{k+1} = \theta_k - \eta(2r_{k+1} \odot v_k + \lambda\theta_k)$ (update parameters with weight decay)

end for

return θ_K

B.2. Convolutional Neural Networks. The initial set of step sizes used for each algorithm are
SGDM: {0.05, 0.1, 0.2, 0.3},
Adam: {1e-4, 3e-4, 5e-4, 1e-3, 2e-3},
AdamW: {5e-4, 1e-3, 3e-3, 5e-3},
AEGD: {0.1, 0.2, 0.3, 0.4},
AEGDW: {0.6, 0.7, 0.8, 0.9}.

The results presented in Figure 6 show that Adam(W) does not generalize as well as SGDM and AEGD(W). Therefore, we only compare the generalization capability of AEGD(W) with SGDM for the remainder of the experiments.

MLP on MNIST. We train a simple multi-layer perceptron (MLP), a special class of feedforward neural networks, with one hidden layer of 200 neurons for the multi-class classification problem on MNIST data set. We run 50 epochs with a batch size of 128 and a weight decay of 10^{-4} for this experiment. Figure 7 (b) shows that AEGDW performs slightly better than SGDM in this case. This is as expected for simple networks.

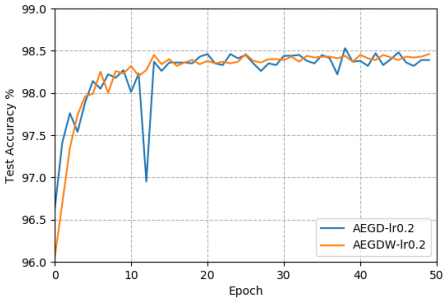
CifarNet on CIFAR-10. We also train a simple 3-block convolutional neural network and name it CifarNet on CIFAR-10. We use the same training set as before – that is, reduce the learning rates by 10 after 150 epochs – with a minibatch size of 128 and weight decay of 10^{-4}. Results for this experiment are reported in Figure 7 (d). The overall performance of each algorithm for CifarNet on CIFAR-10 is similar to the experiments in Figure 6.

AEGDW as an improved AEGD can help to reduce the variance and generalize better early, but such generalization performance does not seem to sustain after decaying the learning rate as pre-scheduled (see examples in Figure 7 (a) and (c)). Overall, AEGDW or AEGD can give better generalization performance than SGDM as evidenced by our experiments.

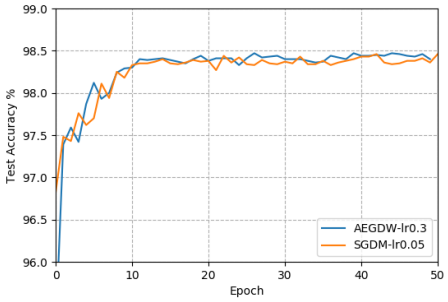
Data availability: The data that support the findings of this study are publicly available online at http://yann.lecun.com/exdb/mnist/ and https://www.cs.toronto.edu/~kriz/cifar.html.

REFERENCES

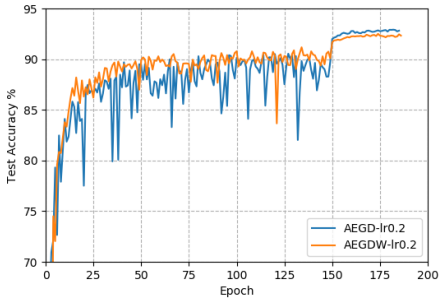
[1] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, *Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz*



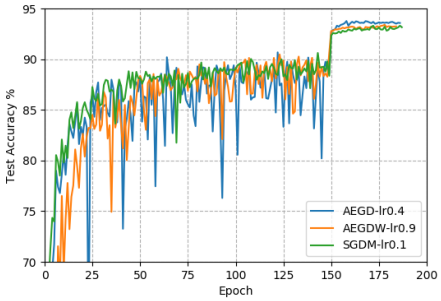
(a) AEGD v.s. AEGDW



(b) Test Accuracy for MLP on MNIST



(c) AEGD v.s. AEGDW



(d) Test Accuracy for CifarNet on CIFAR-10

Fig. 7: Test accuracy for MLP on MNIST and CifarNet on CIFAR-10

inequality, Math. Oper. Res., 35 (2010), pp. 438–457.

[2] H. Attouch, J. Bolte, and B. Svaiter, *Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods*, Mathematical Programming, 137 (2013), pp. 91–129.

[3] H. Attouch, Z. Chbani, J. Fadili, and H. Riahi, *First-order optimization algorithms via inertial systems with hessian driven damping*, Mathematical Programming, (2020), pp. 1–43.

[4] J. Barzilai and J. M. Borwein, *Two-point step size gradient methods*, IMA journal of numerical analysis, 8 (1988), pp. 141–148.

[5] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM journal on imaging sciences, 2 (2009), pp. 183–202.

[6] J. Bolte, A. Daniilidis, O. Ley, and L. Mazet, *Characterizations of łojasiewicz inequalities: Subgradient flows, talweg, convexity*, Transactions of The American Mathematical Society - TRANS AMER MATH SOC, 362 (2009), pp. 3319–3363.

[7] J. Bolte, S. Sabach, and M. Teboulle, *Proximal alternating linearized minimization for nonconvex and nonsmooth problems*, Mathematical Programming, 146 (2014), pp. 459–494.

[8] S. Bos and E. Chug, *Using weight decay to optimize the generalization ability of a perceptron*, in Proceedings of International Conference on Neural Networks, vol. 1, 1996, pp. 241–246.

[9] L. Bottou, *Stochastic gradient descent tricks*, Neural Networks: Tricks of the Trade, 7700 (2012), pp. 421–436.

[10] L. Bottou and Y. Bengio, *Convergence properties of the k-means algorithms*, Advances in neural information processing systems, 7 (1994).

[11] L. Bottou, F. E. Curtis, and J. Nocedal, *Optimization methods for large-scale machine learning*, SIAM Review, 60(2) (2018), pp. 223–311.

[12] C. Castera, J. Bolte, C. Févotte, and E. Pauwels, *An inertial newton algorithm for deep learning.*, J. Mach. Learn. Res., 22 (2021), pp. 134–1.

[13] A. Defazio, F. Bach, and S. Lacoste-Julien, *SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives*, Advances in neural information processing systems, 27 (2014).
[14] D. Di Serafino, V. Ruggiero, G. Toraldo, and L. Zanni, *On the steplength selection in gradient methods for unconstrained optimization*, Applied Mathematics and Computation, 318 (2018), pp. 176–195.
[15] J. Duchi, E. Hazan, and Y. Singer, *Adaptive subgradient methods for online learning and stochastic optimization*, Journal of Machine Learning Research, 12(61) (2011), pp. 2121–2159.
[16] S. Ghadimi and G. Lan, *Accelerated gradient methods for nonconvex nonlinear and stochastic programming*, Mathematical Programming, 156 (2016), pp. 59–99.
[17] P. Gong, C. Zhang, Z. Lu, J. Huang, and J. Ye, *A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems*, in international conference on machine learning, 2013, pp. 37–45.
[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
[19] R. M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtárik, *SGD: General analysis and improved rates*, in International Conference on Machine Learning, 2019, pp. 5200–5209.
[20] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[21] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size*, arXiv preprint arXiv:1602.07360, (2016).
[22] C. Jin, P. Netrapalli, and M. I. Jordan, *Accelerated gradient descent escapes saddle points faster than gradient descent*, in Proceedings of Machine Learning Research, vol. 75, 2018, pp. 1042–1085.
[23] R. Johnson and T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, in Advances in Neural Information Processing Systems 26, 2013, pp. 315–323.
[24] H. Karimi, J. Nutini, and M. Schmidt, *Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition*, in Joint European conference on machine learning and knowledge discovery in databases, Springer, 2016, pp. 795–811.
[25] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, (2014).
[26] A. Krogh and J. A. Hertz, *A simple weight decay can improve generalization*, in Advances in Neural Information Processing Systems 4, 1992, pp. 950–957.
[27] K. Kurdyka, *On gradients of functions definable in o-minimal structures*, Annales de l'Institut Fourier, 48 (1998), pp. 769–783.
[28] L. Lei, C. Ju, J. Chen, and M. I. Jordan, *Non-convex finite-sum optimization via SCSG methods*, in Advances in Neural Information Processing Systems 30, 2017, pp. 2348–2358.
[29] H. Li and Z. Lin, *Accelerated proximal gradient methods for nonconvex programming*, Advances in neural information processing systems, 28 (2015).
[30] H. Liu and X. Tian, *SGEM: stochastic gradient with energy and momentum*, arXiv preprint arXiv:2208.02208, (2022).
[31] H. Liu and P. Yin, *Unconditionally energy stable DG schemes for the Swift-Hohenberg equation*, Journal of Scientific Computing, 81 (2019), pp. 789–819.
[32] H. Liu and P. Yin, *Unconditionally energy stable discontinuous galerkin schemes for the cahn–hilliard equation*, Journal of Computational and Applied Mathematics, 390 (2021), p. 113375.
[33] S. P. Lloyd, *Least squares quantization in PCM*, IEEE Transactions on Information Theory, 28 (1982), pp. 129–137.
[34] S. Lojasiewicz, *A topological property of real analytic subsets*, Coll. du CNRS, Les équations aux dérivées partielles, 117 (1963), p. 2.
[35] I. Loshchilov and F. Hutter, *Decoupled weight decay regularization*, arXiv preprint arXiv:1711.05101, (2017).
[36] L. Luo, Y. Xiong, Y. Liu, and X. Sun, *Adaptive gradient methods with dynamic bound of learning rate*, in International Conference on Learning Representations, 2018.
[37] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, vol. 87, Springer Science & Business Media, 2003.
[38] Y. Nesterov, *Gradient methods for minimizing composite functions*, Mathematical programming, 140 (2013), pp. 125–161.
[39] Y. Nesterov, A. Gasnikov, S. Guminov, and P. Dvurechensky, *Primal–dual accelerated*

gradient methods with small-dimensional relaxation oracle, Optimization Methods and Software, 36 (2021), pp. 773–810.

[40] Y. E. Nesterov, *A method for solving the convex programming problem with convergence rate $o(1/k^2)$*, Dokl. Akad. Nauk SSSR, 269 (1983), pp. 543–547.

[41] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 2006.

[42] P. Ochs, Y. Chen, T. Brox, and T. Pock, *ipiano: Inertial proximal algorithm for nonconvex optimization*, SIAM Journal on Imaging Sciences, 7 (2014), pp. 1388–1419.

[43] S. Osher, B. Wang, P. Yin, X. Luo, F. Barekat, M. Pham, and A. Lin, *Laplacian smoothing gradient descent*, Research in the Mathematical Sciences, 9 (2022), pp. 1–26.

[44] B. Polyak, *Some methods of speeding up the convergence of iteration methods*, USSR Computational Mathematics and Mathematical Physics, 4 (1964), pp. 1–17.

[45] B. Polyak and A. Juditsky, *Acceleration of stochastic approximation by averaging*, SIAM Journal on Control and Optimization, 30 (1992), pp. 838–855.

[46] B. T. Polyak, *Gradient methods for minimizing functionals*, Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, 3 (1963), pp. 643–653.

[47] N. Qian, *On the momentum term in gradient descent learning algorithms*, Neural Networks, 12 (1999), pp. 145–151.

[48] H. Robbins and S. Monro, *A stochastic approximation method*, The Annals of Mathematical Statistics, 22 (1951), pp. 400–407.

[49] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM Journal on Control and Optimization, 14 (1976), pp. 877–898.

[50] J. R. Sashank, K. Satyen, and K. Sanjiv, *On the convergence of adam and beyond*, in International Conference on Learning Representations, vol. 5, 2018, p. 7.

[51] O. Shamir and T. Zhang, *Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes*, in International conference on machine learning, 2013, pp. 71–79.

[52] A. Shapiro and Y. Wardi, *Convergence analysis of gradient descent stochastic algorithms*, Journal of Optimization Theory and Applications, 91 (1996), pp. 439–454.

[53] J. Shen, J. Xu, and J. Yang, *A new class of efficient and robust energy stable schemes for gradient flows*, SIAM Review, 61 (2019), pp. 474–506.

[54] 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 27, 2014, pp. 2510–2518.

[55] T. Tieleman and G. Hinton, *Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning*, Technical report, (2017).

[56] B. Wang, D. Zou, Q. Gu, and S. J. Osher, *Laplacian smoothing stochastic gradient markov chain monte carlo*, SIAM Journal on Scientific Computing, 43 (2021), pp. A26–A53.

[57] X. Yang, *Linear, first and second-order, unconditionally energy stable numerical schemes for the phase field model of homopolymer blends*, Journal of Computational Physics, 327 (2016), pp. 294–316.

[58] P. Yin, M. Pham, A. Oberman, and S. Osher, *Stochastic backward Euler: an implicit gradient descent algorithm for K-means clustering*, Journal of Scientific Computing, 77 (2018), pp. 1133–1146.

[59] G. Zhang, C. Wang, B. Xu, and R. Grosse, *Three mechanisms of weight decay regularization*, in International Conference on Learning Representations, 2018.

[60] H. Zhang and W. W. Hager, *A nonmonotone line search technique and its application to unconstrained optimization*, SIAM journal on Optimization, 14 (2004), pp. 1043–1056.

[61] S. Zhang, A. E. Choromanska, and Y. LeCun, *Deep learning with Elastic Averaging SGD*, in Advances in Neural Information Processing Systems 28, 2015, pp. 685–693.

[62] J. Zhao, Q. Wang, and X. Yang, *Numerical approximations for a phase field dendritic crystal growth model based on the invariant energy quadratization approach*, International Journal for Numerical Methods in Engineering, 110 (2017), pp. 279–300.

[63] Z. Zhu, *Katyusha: The first direct acceleration of stochastic gradient methods*, The Journal of Machine Learning Research, 18(1) (2018), pp. 1–51.