GRADIENT PRECONDITIONING FOR NON-LIPSCHITZ SMOOTH NONCONVEX OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

First-order optimization methods often perform poorly on non-Lipschitz smooth and ill-conditioned problems. Recent work introduced the dual preconditioned gradient descent algorithm, which applies nonlinear preconditioning to the gradient map to improve performance on convex functions satisfying relative smoothness – a generalized version of Lipschitz gradient smoothness. In this paper, we significantly extend this prior work by providing a convergence analysis of this algorithm for non-Lipschitz smooth nonconvex problems. To this end, we exploit recent connections with generalized versions of convexity and smoothness, such as anisotropic convexity and smoothness, which guarantee convergence to a first-order stationary point. Further, we show that some recently proposed preconditioners based on power functions or relativistic dynamics are well-suited for a broad class of objectives. Our experiments demonstrate improved performance using these preconditioners on a variety of non-Lipschitz smooth, nonconvex optimization objectives, including large-scale deep learning tasks.

1 INTRODUCTION

Decades of research have resulted in a rich theory of first-order methods in optimization. Underpinned by this theory, first-order algorithms like gradient descent (GD), classical momentum (Polyak, 1964), and Adam (Kingma & Ba, 2014) have proven practical and effective on a wide variety of optimization tasks, including deep learning. However, standard convergence results require global Lipschitz smoothness, which is often violated in real-world problems. Fast optimization on functions with high and varying curvature has motivated work on momentum-based methods and algorithms with adaptive stepsizes. These methods often improve empirical performance and sometimes enjoy faster theoretical rates, but they still lack convergence guarantees on problems that do not satisfy global Lipschitz smoothness.

A recent body of work (Bauschke et al., 2017; Lu et al., 2018; Bolte et al., 2018; Maddison et al., 2018; 2021) proposes the use of *nonlinear* preconditioning to extend convergence guarantees beyond this set of Lipschitz smooth functions. When solving ill-conditioned linear systems $Ax = b$, it is common to apply a positive definite preconditioning matrix P to reduce the condition number and speed up convergence of first-order algorithms. Similarly, nonlinear preconditioning methods apply a convex nonlinear map that complements the underlying geometry of the optimization problem, where the quality of this match determines the convergence properties of the algorithm.

The most well known nonlinear preconditioning method is Mirror Descent (MD), which can be interpreted as applying a nonlinear preconditioning map to the iterates *before* taking the gradient (Nemirovskij & Yudin, 1983; Beck & Teboulle, 2003). Unfortunately, the practicality of Mirror Descent has been limited by the difficulty of finding the appropriate preconditioning function. Recently, Maddison et al. (2021) proposed Dual Preconditioned Gradient Descent (DPGD), where the nonlinear map is applied *after* taking the gradient, which lives in the dual space. For many common optimization objectives, the relationship between the gradient $\nabla f(x)$ and the local curvature of $f(x)$ is simpler than for the iterates x.[1]

[1]A key example throughout this paper: for objectives behaving like polynomials of high degree, large gradients mean we are in the tails, where there is high curvature. However, inferring whether we are in the tails from iterates alone requires knowledge of where the minimum $x_\star$ is.

The existing analysis of DPGD from Maddison et al. (2021) establishes convergence guarantees for convex functions under *dual relative smoothness* and *dual relative strong convexity*, which are generalizations of Lipschitz smoothness and strong convexity. However, it lacks non-convex guarantees and focuses on simple objectives, not addressing many of the objective functions that arise in real-world applications such as deep learning tasks.

Contributions. In this paper, we aim to provide deeper theoretical insight into DPGD, particularly in the non-convex case. In addition, we focus on questions relevant to practitioners, such as (1) on which problems DPGD may improve performance? and (2) how to choose the appropriate preconditioner for a given problem? Concretely, we provide:

- A non-convex convergence analysis for DPGD based on notions of anisotropic smoothness and convexity recently proposed by Laude et al. (2021).
- Suggested preconditioners for common classes of objective functions. In particular, we show that power preconditioners and the relativistic preconditioner of França et al. (2020) are more robust and performant on functions with fast-growing (super-quadratic) tails, which are ubiquitous in optimization and machine learning.
- Numerical experiments on standard optimization benchmarks as well as large-scale deep learning tasks that support our theoretical findings and demonstrate improved performance of our suggested preconditioners.

2 Background

In this section, we review background material that will be needed for our analysis of DPGD, including various properties such as relative smoothness and relative strong convexity. We will focus on the problem of unconstrained minimization of a differentiable function

$$\min_{x \in \mathbb{R}^d} f(x). \tag{1}$$

We will refer to any $x_\star \in \operatorname{argmin}_{x \in \mathbb{R}^d} f(x)$ as a solution to equation 1.

Definition 2.1 (Bregman Divergence)**.** The Bregman divergence of f for points $x, y \in \mathcal{X}$ is the difference between the value of f at point y and its first order Taylor series approximation at a point x,

$$D_f(y, x) = f(y) - f(x) - \langle \nabla f(x), y - x \rangle. \tag{2}$$

2.1 Dual Preconditioned Gradient Descent

Algorithm Let the preconditioner $k : \mathbb{R}^d \to \mathbb{R}^+$ be a Legendre[2] convex function that is minimized at 0, i.e. $\nabla k(0) = 0$ and $k(0) = 0$. The DPGD update for a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is given by:

$$x_{i+1} = x_i - \frac{1}{L^*} \nabla k(\nabla f(x_i)), \tag{3}$$

where $i \geq 0$, $x_0 \in \mathbb{R}^d$, and $L^* > 0$.

Note that we choose $k(p) = \frac{1}{2}||p||^2$, we obtain the Gradient Descent (GD) update.

Maddison et al. (2021) propose two interpretations of DPGD. The first is as a generalization of left linear preconditioning of gradient descent in which the gradient ∇f is preconditioned by the gradient map ∇k. The second interpretation is as a Mirror Descent step in the dual space of gradients, minimizing the objective $k(\nabla f(x))$ and using f^* (the convex conjugate of f) as the preconditioner. Based on this observation, Maddison et al. (2021) develop a dual theory of convergence for DPGD, borrowing ideas from recent works on MD by Bolte et al. (2018); Bauschke et al. (2017); Lu et al. (2018). In particular, these prior works generalized the notions of smoothness and strong convexity to be defined relative to the preconditioner and showed that the rates of convergence of mirror descent are preserved under these generalized conditions. They referred to these conditions as *relative smoothness* and *relative strong convexity*, which are defined as follows.

[2] $k : \mathbb{R}^d \to \mathbb{R}$ is Legendre convex if it is differentiable, strictly convex, and has $\lim_{||x|| \to \infty} ||\nabla k(x)|| \to \infty$.

Definition 2.2 (Relative smoothness and relative strong convexity (Bauschke et al., 2017; Lu et al., 2018))**.** Let f, h be differentiable convex functions defined on $\mathbb{R}^d$ and $0 \leq \mu \leq L$,

1. f is L-smooth relative to h on $\mathbb{R}^d$ if $Lh - f$ is convex, or equivalently, for any $x, y \in \mathbb{R}^d$,
$$D_f(y, x) \leq L D_h(y, x). \tag{4}$$

2. f is μ-strongly convex relative to h on $\mathbb{R}^d$ if $f - \mu h$ is convex, or equivalently, for any $x, y \in \mathbb{R}^d$,
$$\mu D_h(y, x) \leq D_f(y, x). \tag{5}$$

If $h(x) = \frac{1}{2}||x||^2$, we recover classical Lipschitz-smoothness and strong convexity,

$$\frac{\mu}{2}||y - x||^2 \leq f(y) - f(x) - \langle \nabla f(x), y - x \rangle \leq \frac{L}{2}||y - x||^2 \quad \forall\, x, y \in \mathbb{R}^d. \tag{6}$$

These relative conditions are also referred to as *primal* relative smoothness and *primal* relative strong convexity, in constrast with *dual* relative smoothness and *dual* relative strong convexity introduced by Maddison et al. (2021).

Definition 2.3 (Dual relative smoothness and dual relative strong convexity Maddison et al. (2021))**.** Let $f, k : \mathbb{R}^d \to \mathbb{R}$ be differentiable, k be convex, and f be strongly convex with convex conjugate f^*. We say that

1. k is L^*-smooth relative to f^* on $\mathbb{R}^d$ if for any $x, y \in \mathbb{R}^d$,
$$D_k(\nabla f(y), \nabla f(x)) \leq L^* D_{f^*}(\nabla f(y), \nabla f(x)). \tag{7}$$

2. k is μ^*-strongly convex relative to f^* on $\mathbb{R}^d$ if for any $x, y \in \mathbb{R}^d$,
$$\mu^* D_{f^*}(\nabla f(y), \nabla f(x)) \leq D_k(\nabla f(y), \nabla f(x)) \tag{8}$$

Convergence rates for DPGD on convex functions. Maddison et al. (2021) provided convergence rates for DPGD on differentiable convex functions f with minimum $x_\star$. Under L^*-smoothness relative to k, they proved that for all $i > 0$, the iterates of DPGD satisfy

$$k(\nabla f(x_i)) \leq \frac{L^*}{i}(f(x_0) - f(x_\star)). \tag{9}$$

If additionally f is Legendre[2] and μ^*-strongly convex relative to k, then the iterates satisfy

$$f(x_i) - f(x_\star) \leq (1 - \frac{\mu^*}{L^*})^i (f(x_0) - f(x_\star)). \tag{10}$$

Unfortunately, the above conditions and convergence results rely heavily on the strict convexity of f, which allows for transformations between inequalities on f^* (dual relative conditions) to inequalities on f (convergence bounds). For a non-convex analysis of DPGD, we seek dual relative smoothness inequalities that upper bound f rather than f^*.

2.2 Anisotropic Smoothness and Anisotropic Convexity

In the Euclidean case, there is a duality between Lipschitz smoothness and strong convexity in the sense that if f is Lipschitz smooth, then f^* is strongly convex and vice versa. While relative smoothness and relative strong convexity generalize Lipschitz smoothness and strong convexity to Bregman divergences, they do not preserve the property that if f is smooth relative to h then f^* is strongly convex relative to h^*. Laude et al. (2021) address this issue by introducing notions of anisotropic strong convexity and smoothness as the respective dual counterparts of relative smoothness and strong convexity, leveraging rich ideas from generalized convexity (Dolecki & Kurcyusz, 1978; Rockafellar & Wets, 2009).

In this section, we restate some of the key results and definitions that will be necessary to establish our convergence results and refer the reader to Laude et al. (2021) for details. Note that the terminology a-weak/a-strong convexity and a-smoothness is used as a shorthand for anisotropic weak/strong convexity and anisotropic smoothness.

Definition 2.4. (Laude et al. (2021), Definition 3.6) Let $\phi : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable, coercive, and strictly convex. And let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable.

1. f is **a-weakly convex** with respect to ϕ if for any $x, y \in \mathbb{R}^d$
$$f(y) \geq f(x) - \phi(y - x + \nabla\phi^*(-\nabla f(x))) + \phi(\nabla\phi^*(-\nabla f(x)). \tag{11}$$
2. f is **a-strongly convex** with respect to ϕ if for any $x, y \in \mathbb{R}^d$
$$f(y) \geq f(x) + \phi(y - x + \nabla\phi^*(\nabla f(x))) - \phi(\nabla\phi^*(\nabla f(x)). \tag{12}$$
3. f satisfies the **anisotropic descent lemma** if $-f$ is a-weakly convex, or equivalently if for every $x \in \mathbb{R}^n$ it holds that
$$f(y) \leq f(x) + \phi(y - x + \nabla\phi^*(\nabla f(x)) - \phi(\nabla\phi^*(\nabla f(x))) \quad \forall y \in \mathbb{R}^n. \tag{13}$$
4. f is **a-smooth** if f is continuously differentiable and both f and $-f$ are a-weakly convex.

Remark 2.5. For convex functions, a-smoothness is equivalent to the anisotropic descent lemma.

Remark 2.6. For strictly convex functions, dual relative smoothness defined by Maddison et al. (2021) is equivalent to the a-weak convexity of $-f$ or to the anisotropic descent lemma. Anisotropic smoothness can thus be seen as a generalization of dual relative smoothness.

3 Convergence of DPGD on nonconvex functions

In this section, we use the anisotropic descent lemma to establish a convergence rate for DPGD on nonconvex functions. First, recall that the gradient descent update can be rewritten as follows

$$x_{i+1} = \arg\min_x f(x_i) + \nabla f(x_i)(x - x_i) + \frac{1}{2L}||x - x_i||^2, \tag{14}$$

which can be interpreted as a minimization of the quadratic upper bound derived from Lipschitz smoothness in equation 6

$$f(x_{i+1}) \leq f(x_i) + \nabla f(x_i)(x_{i+1} - x_i) + \frac{1}{2L}||x_{i+1} - x_i||^2. \tag{15}$$

Similarly, we consider the nonlinear upper bound provided by the anisotropic descent lemma and define the following update where we have chosen $\phi(p) = \frac{1}{L^*}k^*(L^*p)$

$$x_{i+1} = \arg\min_x f(x_i) + \frac{1}{L^*}k^*(L^*(x - x_i + \frac{1}{L^*}\nabla k(\nabla f(x_i)))) - \frac{1}{L^*}k^*(\nabla k(\nabla f(x_i))), \tag{16}$$

The RHS is minimized when its gradient vanishes: $\nabla k^*(L^*(x_{i+1} - x_i + \frac{1}{L^*}\nabla k(\nabla f(x_i)))) = 0$, and by composition with the invertible map ∇k, it is equivalent to

$$x_{i+1} - x_i + \frac{1}{L^*}\nabla k(\nabla f(x_i)) = \frac{1}{L^*}\nabla k(0). \tag{17}$$

Thus allowing us to recover the DPGD update $x_{i+1} = x_i - \frac{1}{L^*}\nabla k(\nabla f(x_i))$ since $\nabla k(0) = 0$. This shows that the anisotropic descent lemma generalizes the classical descent lemma used to establish convergence rates for GD. Further, we know that we can guarantee a sufficient decrease at each iteration as long as

$$f(x_{i+1}) \leq f(x_i) + \frac{1}{L^*}k^*(L^*(x_{i+1} - x_i + \frac{1}{L^*}\nabla k(\nabla f(x_i)))) - \frac{1}{L^*}k^*(\nabla k(\nabla f(x_i))) \tag{18}$$

which is equivalent to

$$f(x_{i+1}) - f(x_i) \leq \frac{1}{L^*}k^*(0) - \frac{1}{L^*}k^*(\nabla k(\nabla f(x_i))) = -\frac{1}{L^*}k^*(\nabla k(\nabla f(x_i))) \leq 0. \tag{19}$$

Clearly, if f satisfies the **anisotropic descent lemma** with respect to the preconditioner, then condition 18 holds not only for convex functions but for nonconvex functions as well. Moreover, the last inequality can be interpreted as a generalization of the descent condition used to prove the convergence of Gradient Descent on Lipschitz smooth functions, which is given by

$$f(x_{i+1}) - f(x_i) \leq -\frac{1}{2L}||\nabla f(x_i)||^2 \quad \forall\, i \geq 0, \tag{20}$$

where L is the Lipschitz constant for the gradient of f. It also generalizes the descent condition for the Rescaled Gradient Descent algorithm of order M introduced by Wilson et al. (2019), which is itself a special case of DPGD,

$$f(x_{i+1}) - f(x_i) \leq -\eta||\nabla f(x_i)||^{\frac{M}{M-1}} \quad \forall\, i \geq 0. \tag{21}$$

where $\eta > 0$.

We formalize this assumption in the following definition.

Definition 3.1. Let $f, k : \mathbb{R}^d \to \mathbb{R}$ with k Legendre convex and minimized at $k(0) = 0$. Then DPGD is said to satisfy the **generalized descent condition** for the pair (f, k) if there exists a constant $\eta > 0$ such that for all iterates $x_i \in \mathbb{R}^d$ we have

$$f(x_{i+1}) - f(x_i) \leq -\eta k^*(\nabla k(\nabla f(x_i))). \tag{22}$$

Note that $k^*(\nabla k(\nabla f(x_i))) = D_k(0, \nabla f(x_i))$ where D_k is the Bregman divergence of k and the RHS of the generalized descent condition can thus be interpreted as a measure of the distance between $\nabla f(x_i)$ and 0. This drives the convergence of the sequence of iterates x_i as the gradient gets close to 0, i.e. as we approach a first order stationary point. Based on this observation, we can obtain a convergence rate for nonconvex functions.

Theorem 3.2. *Let $f, k : \mathbb{R}^n \to \mathbb{R}$ with k Legendre convex and minimized at $k(0) = 0$. If the pair (f, k) satisfies the generalized descent condition, then DPGD converges to a first-order stationary point with the rate*

$$\min_{0 \leq i \leq T} D_k(0, \nabla f(x_i)) \leq \frac{f(x_0) - f(x_\star)}{\eta(T+1)}. \tag{23}$$

The proof of Theorem 3.2 is in Appendix A.1.

Remark 3.3. Theorem 3.2 only assumes the generalized descent condition and not the anisotropic descent lemma because the latter is a consequence of the former. In fact, one can see that the anisotropic descent lemma is a global condition that must be satisfied for any two points $x, y \in \mathbb{R}^n$, whereas the generalized descent condition only constrains consecutive iterates.

Remark 3.4. The convergence guarantee provided in Theorem 3.2 is for the quantity $D_k(0, \nabla f(x_i))$ whereas a common quantity of interest is $||\nabla f(x_i)||$ in the nonconvex case. This rate can be recovered if $k(p) = g(||p||)$ is a function of the norm, as is the case for the preconditioners we consider in Section 4. For example, for Gradient descent, the preconditioner map is $k(p) = \frac{1}{2}||p||^2$, so equation 23 yields the familiar rate of non-convex convergence

$$\min_{0 \leq i \leq T} ||\nabla f(x_i)|| \leq \sqrt{\frac{2L(f(x_0) - f(x_\star))}{(T+1)}}. \tag{24}$$

Similarly, the Rescaled Gradient Descent algorithm of order M introduced by Wilson et al. (2019) corresponds to $k(p) = ||p||^{\frac{M}{M-1}}$. Its convergence rate can also be recovered from equation 23:

$$\min_{0 \leq i \leq T} ||\nabla f(x_i)|| \leq \Big(\frac{f(x_0) - f(x_\star)}{\eta(T+1)}\Big)^{\frac{M-1}{M}}. \tag{25}$$

4 Pairing Preconditioners and Objective Functions

The choice of the preconditioner k in DPGD is critical to its success: the more accurately k reflects the dual-space geometry of f, the better DPGD performs. The theory of convex duality says that if f is a Legendre convex function, then $\nabla f(x)$ is an invertible map relating coordinates in the primal and dual spaces with the special property that $\nabla f^* = (\nabla f)^{-1}$. Leveraging this idea, we obtain that the ideal preconditioner is f_c^* where f_c is the centered version of f defined by

$$f_c(x) = f(x + x_\star) - f(x_\star), \quad \forall x \in \mathbb{R}^d. \tag{26}$$

In this case, we have $\nabla f_c^*(\nabla f(x)) = \nabla f^*(\nabla f(x)) - x_\star = x - x_\star$. Thus, DPGD with $k = f_c^*$ and $L^* = 1$ converges in one iteration. However, in general, computing the convex conjugate of f_c is as difficult as minimizing f, so we must relax the condition $k = f_c$ by considering functions that approximately match the dual-space geometry of f, based on our prior knowledge. This approximate matching is formalized in the condition of anisotropic smoothness/the anisotropic descent lemma.

4.1 Polynomials and Power Preconditioners

We wish to choose models for f that well-approximate its geometry but also have an easy-to-compute convex conjugate k. One such powerful group of geometric models is the class of power functions $f(x) = \frac{1}{M}||x||^M$ for $x \in \mathbb{R}^d$ and $M > 1$. This is due to the fact that the Fenchel dual of f_c can be expressed analytically as $f_c^*(p) = \frac{M-1}{M}||p||^{\frac{M}{M-1}}$ for $p \in \mathbb{R}^d$. Note that the Rescaled Gradient Descent algorithm implicitly uses the preconditioner $k(p) = \frac{M-1}{M}||p||^{\frac{M}{M-1}}$ and is thus well suited for functions that behave like power functions of order M.

Maddison et al. (2018) extend this idea to convex functions with distinct growth behaviours near the minimum (body) and far from the minimum (tail). For example, a function may grow like $||x - x_\star||^b$ near its minimum and $||x - x_\star||^B$ at the tail with $b \neq B$. In this case, one can consider any preconditioner that interpolates between the functions $||p||^a$ for p near 0 and $||p||^A$ away from 0, where $a = \frac{b}{b-1}$ and $A = \frac{B}{B-1}$. We will use this concept of power growth and interpolating preconditioners to build towards an analysis of general nonconvex functions. In particular, distinct growth behaviors in the body and tail is a property of polynomial objectives, which encompass or resemble a wide variety of optimization objectives (from toy to deep learning objectives). The next result provides a matching preconditioner for the family of univariate polynomial functions.

Theorem 4.1. *Consider a polynomial $f(x) = \sum_{i=0}^n a_i x^i$, with $n \geq 2$. Then DPGD with*

$$k(p) = \begin{cases} \frac{1}{2}|p|^2 & \text{if } |p| \leq 1, \\ \frac{N-1}{N}|p|^{\frac{N}{N-1}} + \frac{1}{N} - \frac{1}{2} & \text{otherwise}, \end{cases} \tag{27}$$

where $N \geq n$, satisfies the generalized descent condition.

The proof of this result is in Appendix A.1. This result holds for any Legendre preconditioner k such that $k(p) \approx \frac{1}{2}|p|^2$ when $|p| \ll 1$ and $k(p) \approx \frac{N-1}{N}|p|^{\frac{N}{N-1}}$ when $|p| \gg 1$. Without prior knowledge about the degree of f, Theorem 4.1 suggests that one should choose N to be as large as possible and ideally consider $N \to \infty$. One way to achieve this interpolation is with a relativistic preconditioner $k(p) = \sqrt{||p||^2 + 1} - 1$. More generally, and borrowing from Maddison et al. (2018), we can define a class of preconditioners that interpolates between the functions $||p||^a$ for p near 0 and $||p||^A$ away from 0 as follows

$$k(p) = \frac{1}{A}(\delta||p||^a + 1)^{\frac{A}{a}} - \frac{1}{A}, \tag{28}$$

where $a, A \geq 1$ and $\delta > 0$. We refer to DPGD with this class of preconditioners as Power Descent (PD). The relativistic preconditioner is thus one particular case where $a = 2$ and $A = 1$. We refer to DPGD with relativistic preconditioner as Relativistic Gradient Descent (RGD).

In general, PD and RGD seek to better *approximate* the dual space geometry of the function, and thus may only satisfy the generalized descent condition locally. However, for functions with roughly polynomial growth, we can expect the generalized descent condition for PD and RGD to be satisfied in a significantly larger region of the parameter space than for GD. For example, consider the simple overparametrized function $f(x, y) = \frac{1}{2}(xy)^2$. For a given step size η and points (x, y), we can check whether the generalized descent condition starting from (x, y) is satisfied or not for different preconditioners (see Figure 1).

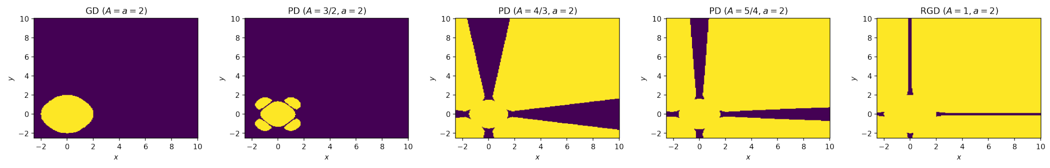


Figure 1: Characterization of the regions where the descent condition is satisfied for GD, PD and RGD on the 4th degree polynomial $f(x, y) = \frac{1}{2}(xy)^2$ with step size = 0.25. The yellow regions indicate where the descent lemma is satisfied and the purple regions indicate where it is not. Note the phase transition at $A = 4/3$, the conjugate power to a 4-th order power function. GD only satisfies the descent condition (and therefore only has convergence guarantees) on small initializations near the origin, while PD and RGD converge over nearly all the parameter space.

Figure 1 shows that as the value of A decreases from 2 to 1, the size of the region where we can guarantee a sufficient decrease increases significantly. One might argue that we should always choose $A = 1$ since the relativistic preconditioner maximizes the desired region of descent. However, it is important to note that by doing so, we sacrifice the faster convergence rate guaranteed by PD (see 23). RGD performs a conservative update, which is always smaller than that of PD or GD. So while it is more robust to the choice of step size or initialization, other methods like PD may achieve faster convergence, especially when one can match the power behavior of the function f.

Another class of functions that highlights the benefits of using PD over GD is the class of positively homogeneous functions. Consider a function $f : \mathbb{R}^n \to \mathbb{R}$ that is positively homogeneous of degree i. We thus have $f(cx) = c^i f(x)$ and $\nabla f(cx) = c^{i-1}\nabla f(x)$ for all $x \in \mathbb{R}^n$. We assume that $i > 2$ and we define the generalized descent function at a point $x \in \mathbb{R}^n$ with step size $\eta > 0$:

$$g(x, \eta) = f(x - \eta \nabla k(\nabla f(x))) - f(x) + \eta D_k(0, \nabla f(x)) \tag{29}$$

If $g(x, \eta) \leq 0$, then the generalized descent condition is satisfied between consecutive iterates x and $x - \eta\nabla k(\nabla f(x)$. And we observe the following:

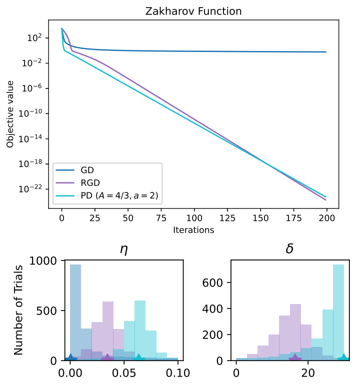
- For GD, we have for all $c > 0$, $g(cx, \eta) = g(x, \eta c^{i-2})$. Therefore, increasing the scaling factor c significantly increases the effective step size or local Lipschitz constant of the function.
- Using the preconditioner k from theorem 4.1 with N=i, we have for all $c > 0$, $g(cx, \eta) = g(x, \eta)$ when $||\nabla f(x)|| \geq 1$ (recall that the algorithm behaves exactly like GD when $||\nabla f(x)|| \leq 1$). This suggests that when we are far from the minimum (large gradients), a power preconditioner is not only faster than GD but is also more robust to the scale of the parameters. This can explain for example the behaviour observed in the deep matrix regression experiment in the next section. GD diverges when the scale of the initialization is large, whereas PD and RGD are able to converge.

5 Experiments

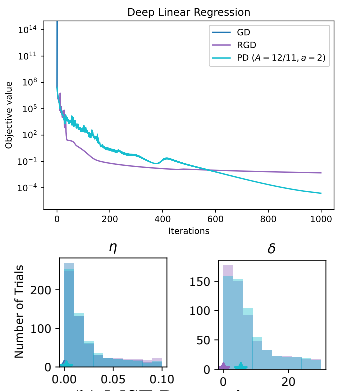
To empirically justify our findings and evaluate the practical performance gains of various preconditioners, we conduct numerical experiments on a set of standard deterministic optimization objectives as well as on large-scale deep learning problems. We consider the gradient descent (GD), power descent (PD), and relativistic gradient descent (RGD) algorithms, their variants with momentum (parameter μ), and the popular adaptive gradient optimizer Adam. We also consider the special case of RGD with momentum discretized by the second order integrator from França et al. (2020), which introduces an additional parameter $\alpha \in [0, 1]$ that interpolates between a Nesterov-style integrator at $\alpha = 0$ and a symplectic leapfrog integrator at $\alpha = 1$. As supported by our theory, we find that RGD exhibits strong performance across a wide range of problems, including deep learning tasks.

5.1 Deterministic Functions

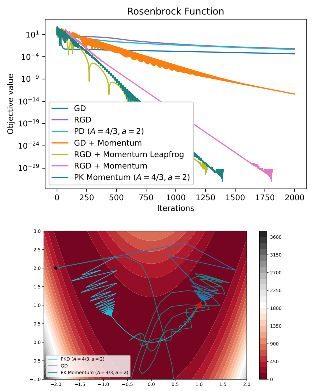
We begin with a set of convex and nonconvex deterministic problems. All hyperparameters were systematically optimized across hundreds of trials using the popular HyperOpt package (Bergstra et al., 2013). By studying the distribution of selected hyperparameters, we can gain an understanding of the sensitivity of each method and determine good cross-task prior values for practitioners.



(a) Convex Zakharov function.



(b) MSE Regression.



(c) 2-D Rosenbrock.

Figure 2: **Zakharov 2a**: Top shows objective value vs iteration for the best hyperparameter setting discovered. Bottom shows the histogram of hyperparameters chosen by Bayesian optimization. RGD and PD are able to minimize this function at a linear rate while GD fails to converge in 200 iterations. **Deep Linear Regression 2b**: with a 6-layer linear network $f(W_1, ..., W_6) = \frac{1}{2}||W_6...W_1X - y||^2$. Gradient descent quickly diverges, RGD and PD are both able to minimize this nonconvex function. **Rosenbrock 2c**: GD, RGD, and PD fail to minimize this function after 2000 iterations. Bottom shows a trajectory plot where the black dot is x_0 and the red star is the optimum $x_\star$. The addition of momentum is crucial - by reducing oscillations in the narrow valley, all momentum-based algorithms converge linearly, although RGD + Momentum and PD + Momentum do so significantly faster.

Zakharov Function. First we consider the convex Zakharov function $f(x) = \sum_{i=1}^{5} x_i^2 + (\frac{1}{2}\sum_{i=0}^{5} ix_i)^2 + (\frac{1}{2}\sum_{i=1}^{5} ix_i)^4$, a 4th order polynomial in 5 dimensions (Figure 2a). Since $f(x)$ is a sum of monomials of degrees 2-4, it has distinct growth behaviors: It behaves like x^4 in the tail and x^2 near the minimum. Thus, we set $A = 4/(4-1) = 3$, $a = 2/(2-1) = 2$ for PD. Both RGD and PD approximate these growth behaviors and converge linearly, while GD does not and fails to converge.

Deep Linear Regression. For a non-convex problem in higher dimensions, we consider MSE regression with a K-layer deep linear network $f(W_1, ..., W_K) = \frac{1}{2}||W_K...W_1X - y||^2$, where $K = 6$. This nonconvex, 600-dimensional, overparametrized problem is intended to reflect features of optimization in deep learning. We initialize y as in the ℓ_4 regression problem and initialize $X, W_1, ..., W_K \in \mathbb{R}^{10\times 10}$ from i.i.d. standard Gaussians. This objective is a $2K = 12$th degree polynomial, informing our choice of $A = 12/(12-1)$ for the PD algorithm.

Note in Figure 2b that gradient descent rapidly diverges with our i.i.d. standard Gaussian initialization for $W_K, ..., W_1$. We found that this no longer occurred if we scaled the standard deviation by a factor of $1/\sqrt{10}$, recovering the popular Xavier initialization used in deep learning Glorot & Bengio (2010). Traditional motivations for such initializations cited issues with "exploding gradients". We explain the effectiveness of these initialization strategies as picking a starting point for gradient descent in the "body" of the loss function, where the objective behaves roughly quadratically.

Rosenbrock Function. Lastly, we consider the 2-D Rosenbrock function $f(x, y) = (1-x)^2 + 100(y-x^2)^2$ (Rosenbrock, 1960), a difficult nonconvex benchmark problem with a long, narrow, flat valley leading to the minimum (Figure 2c). This function has a single global minimum at $x_\star = (1, 1)$ with $f(x_\star) = 0$. We initialize at $x_0 = (-2, 2)$ as in França et al. (2020). Finding the narrow valley is easy, but convergence to the minimum is challenging. Given that the function is a polynomial of degrees 2-4, we set $A = 4/(4-1), a = 2/(2-1) = 2$ for the PD algorithm.

5.2 Deep Learning Experiments

In all experiments, we compared GD + Momentum, RGD + Momentum, RGD + Momentum Leapfrog from França et al. (2020), and Adam. While our theoretical results only analyze algorithms without momentum, we find that in stochastic optimization problems, some form of gradient averaging is crucial in order to obtain low variance estimators of the gradient. This is a common practice – using the bare gradient leads to poor performance.

We tuned hyperparameters with 24 trials of random search. For existing algorithms, we used hyperparameter search spaces informed by similar tasks from Schmidt et al. (2021) and Choi et al. (2019). For the new α, δ parameters in RGD, we set a single reasonable common search space across all experiments (Appendix B). For each task, we show the value of the training objective next to the standard validation metric of interest. In each plot, the solid line represents the best run (as measured by the validation metric) for that algorithm across all trials. The shaded region represents the trajectory of the top 10% of all trials, providing a measure of the sensitivity of the algorithm. We test performance on three problems, image classification on CIFAR100 with a ResNet-34 (He et al., 2016), language modeling on the text of Leo Tolstoy's War and Peace with a character-level LSTM from (Schneider et al., 2019), and image generation on CelebA Faces with a Convolutional VAE (Subramanian, 2020).

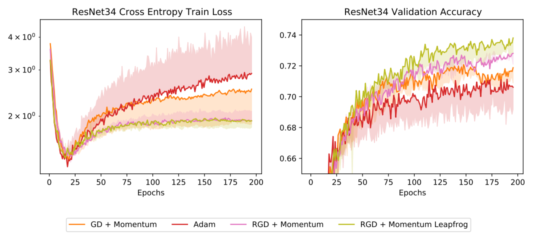


Figure 3: ResNet-34 on CIFAR-100 - Image Classification

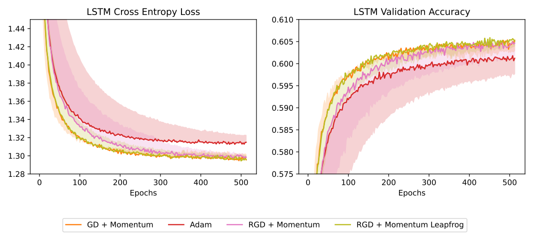


Figure 4: CharLSTM on Tolstoy's War and Peace - Language Modeling

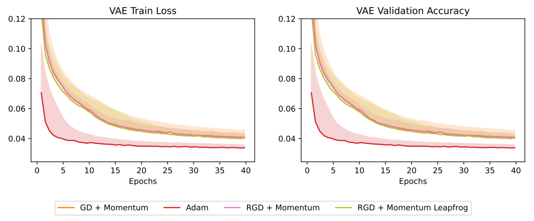


Figure 5: Convolutional VAE on CelebA Faces - Image Generation

We find DPGD with a relativistic preconditioner to be a competitive optimizer on a variety of deep learning tasks. In particular, it outperformed Adam on both supervised learning problems and demonstrated improved or equal performance to SGD + Momentum on all tasks, which it recovers as a special case as $\delta \to 0$. In particular, it exhibits strong performance on the deepest network in our set of tasks – a 32-layer ResNet. Deeper networks tend to be more susceptible to gradient vanishing and explosion, although this has largely been solved through a combination of careful initialization strategies and architectural tricks like residual connections and LSTM gates (He et al., 2016). This suggests that DPGD with the appropriate preconditioner (such as the relativistic preconditioner, which allows for a maximum stepsize of $\eta\sqrt{\delta}$) may allow for a wider range of initializations for deep networks and that it may be stable and effective on problems where the issue of exploding gradients remain, such as in meta-learning or reinforcement learning (Wang et al., 2021). In cases where large gradients are not an issue or have been mitigated through other tricks, we expect algorithms like RGD to behave similarly to gradient descent, as seen in the LSTM and VAE experiments. We leave further exploration of these hypotheses to future work.

6 Conclusion

In this paper, we study the recently proposed dual preconditioned gradient descent algorithm in the nonconvex setting, deriving a natural descent condition that guarantees convergence to a first order stationary point. Leveraging this, we obtain analytical convergence rates across a larger class of non-convex functions than gradient descent. Moreover, we address the issue of matching objective functions with the appropriate preconditioners for provable convergence. In particular, we study preconditioners with power behavior of different orders in the body and tail and demonstrate both theoretically and empirically that they are effective on a wide variety of functions, such as polynomials, deep linear regression, and deep neural networks.

REFERENCES

Heinz H Bauschke, Jérôme Bolte, and Marc Teboulle. A descent lemma beyond lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348, 2017.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pp. 115–123. PMLR, 2013.

Jérôme Bolte, Shoham Sabach, Marc Teboulle, and Yakov Vaisbourd. First order methods beyond convexity and lipschitz gradient continuity with applications to quadratic inverse problems. *SIAM Journal on Optimization*, 28(3):2131–2151, 2018.

Dami Choi, Christopher J Shallue, Zachary Nado, Jaehoon Lee, Chris J Maddison, and George E Dahl. On empirical comparisons of optimizers for deep learning. *arXiv preprint arXiv:1910.05446*, 2019.

Szymon Dolecki and Stanisław Kurcyusz. On ϕ-convexity in extremal problems. *SIAM Journal on Control and Optimization*, 16(2):277–300, 1978.

Guilherme França, Jeremias Sulam, Daniel P Robinson, and René Vidal. Conformal symplectic and relativistic optimization. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12): 124008, 2020.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Emanuel Laude, Andreas Themelis, and Panagiotis Patrinos. Conjugate dualities for relative smoothness and strong convexity under the light of generalized convexity. *arXiv preprint arXiv:2112.08886*, 2021.

Haihao Lu, Robert M Freund, and Yurii Nesterov. Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization*, 28(1):333–354, 2018.

Chris J Maddison, Daniel Paulin, Yee Whye Teh, Brendan O'Donoghue, and Arnaud Doucet. Hamiltonian descent methods. *arXiv preprint arXiv:1809.05042*, 2018.

Chris J Maddison, Daniel Paulin, Yee Whye Teh, and Arnaud Doucet. Dual space preconditioning for gradient descent. *SIAM Journal on Optimization*, 31(1):991–1016, 2021.

Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

HoHo Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184, 1960.

Robin M Schmidt, Frank Schneider, and Philipp Hennig. Descending through a crowded valley-benchmarking deep learning optimizers. In *International Conference on Machine Learning*, pp. 9367–9376. PMLR, 2021.

Frank Schneider, Lukas Balles, and Philipp Hennig. Deepobs: A deep learning optimizer benchmark suite. *arXiv preprint arXiv:1903.05499*, 2019.

A.K Subramanian. Pytorch-vae. `https://github.com/AntixK/PyTorch-VAE`, 2020.

Xiang Wang, Shuai Yuan, Chenwei Wu, and Rong Ge. Guarantees for tuning the step size using a learning-to-learn approach. In *International Conference on Machine Learning*, pp. 10981–10990. PMLR, 2021.

Ashia Wilson, Lester Mackey, and Andre Wibisono. Accelerating rescaled gradient descent: Fast optimization of smooth functions. *arXiv preprint arXiv:1902.08825*, 2019.

A Appendix

A.1 Proof of theorem 3.2

Theorem 3.2. *Let $f, k : \mathbb{R}^n \to \mathbb{R}$ with k Legendre convex and minimized at $k(0) = 0$. If the pair (f, k) satisfies the generalized descent condition, then DPGD converges to a first-order stationary point with the rate*

$$\min_{0 \le i \le T} D_k(0, \nabla f(x_i)) \le \frac{f(x_0) - f(x_\star)}{\eta(T+1)}. \tag{23}$$

Proof. First, we restate the generalized descent condition

$$f(x_{i+1}) - f(x_i) \le -\eta k^*(\nabla k(\nabla f(x_i)) \tag{30}$$

Summing up over iterations in the usual fashion, we obtain

$$\sum_{i=0}^{T} f(x_{i+1}) - f(x_i) \le \sum_{i=0}^{T} -\eta k^*(\nabla k(\nabla f(x_i)), \tag{31}$$

$$\implies f(x_{T+1}) - f(x_0) \le -\sum_{k=0}^{T} {}^*(\nabla k(\nabla f(x_i)), \tag{32}$$

$$\implies f(x_0) - f(x_{T+1}) \ge \sum_{k=0}^{T} \eta k^*(\nabla k(\nabla f(x_i)), \tag{33}$$

$$\implies f(x_0) - f(x_\star) \ge \eta(T+1) \min_{0 \le i \le T} k^*(\nabla k(\nabla f(x_i)), \tag{34}$$

$$\implies \frac{f(x_0) - f(x_\star)}{\eta(T+1)} \ge \min_{0 \le i \le T} k^*(\nabla k(\nabla f(x_i))). \tag{35}$$

And we conclude by noting that

$$\begin{aligned} k^*(\nabla k(\nabla f(x_i))) &= \langle \nabla k(\nabla f(x_i)), \nabla f(x_i) \rangle - k(\nabla f(x_i)) & (36)\\ &= k(0) - k(\nabla f(x_i)) - \langle \nabla k(\nabla f(x_i)), 0 - \nabla f(x_i) \rangle & (37)\\ &= D_k(0, \nabla f(x_i)). & (38) \end{aligned}$$

□

A.2 Proof of theorem 4.1

Theorem 4.1. *Consider a polynomial $f(x) = \sum_{i=0}^{n} a_i x^i$, with $n \ge 2$. Then DPGD with*

$$k(p) = \begin{cases} \frac{1}{2}|p|^2 & \text{if } |p| \le 1, \\ \frac{N-1}{N}|p|^{\frac{N}{N-1}} + \frac{1}{N} - \frac{1}{2} & \text{otherwise}, \end{cases} \tag{27}$$

where $N \ge n$, satisfies the generalized descent condition.

Proof. First, note that since f is a polynomial of degree n then for all $m = 1, \dots, n$, the m-th derivative $f^{(m)}$ is a polynomial of degree $n - m$. In particular, $f^{(n)}$ is a constant function and is everywhere equal to $a_n n!$, and for all $j > n$ the j-th derivative is the zero function.

We will prove that the generalized descent condition is satisfied at any point x_i by considering the two cases $|f'(x_i)| \le 1$ and $|f'(x_i)| > 1$ separately.

Consider the set $A = \{x \in \mathbb{R}; f'(x) = \pm 1\}$. We know that A is finite because the polynomials $f'(x) - 1$ and $f'(x) + 1$ each have a maximum of $n - 1$ roots. Therefore, A has at most $2(n-1)$ elements. We can thus write $A = \{x_1, \dots x_K\}$ such that $x_1 < \dots < x_K$ and $K = \text{card}(A)$. Since $|f'(x)| - 1$ is continuous, it cannot change signs on any interval A_j where

$$A_j = \begin{cases} [x_j, x_{j+1}] & \text{if } i = 1, \dots K-1, \\ (-\infty, x_1] & \text{if } j = 0, \\ [x_N, +\infty) & \text{if } j = K. \end{cases} \tag{39}$$

Note that since $f'(x)$ is a polynomial, we must have $|f'(x)| \geq 1$ on A_0 and A_K. And we can write $\mathbb{R} = \Big(\cup_{j \in J} B_j \Big) \cup \Big(\cup_{j \in J'} B'_j \Big)$ where $(B_j)_{j \in J}$ (resp. $(B'_j)_{j \in J'}$) are the intervals on which we have $|f'(x)| \geq 1$ (resp. $|f'(x)| \leq 1$).

We start by considering the case: $|f'(x_i)| \geq 1$. Recall the descent condition

$$f(x_{i+1}) - f(x_i) \leq -\eta k^*(\nabla k(\nabla f(x_i)) \tag{40}$$

or equivalently,

$$\frac{k^*(\nabla k(\nabla f(x_i))}{f(x_{i+1}) - f(x_i)} \leq \frac{1}{\eta} = \mu \tag{41}$$

On any interval $B_j = [x_j, x_{j+1}]$ where $j \in \{1, \ldots, K-1\}$, the function $g(x_i) = \frac{k^*(\nabla k(\nabla f(x_i))}{f(x_{i+1}) - f(x_i)}$ is continuous and thus bounded by a constant α_j. Moreover, note that $x_{i+1} = \phi(x_i) = x_i - \frac{1}{L^*}|f'(x_i)|^{\frac{1}{N-1}} \sim x_i$ when $x_i \to \pm\infty$. Therefore $f(x_{i+1}) - f(x_i) \sim a x_i^n$ and similarly, we have $k^*(\nabla k(\nabla f(x_i)) \sim b x_i^{\frac{nN}{N-1}}$. And since $N \geq n$, $g(x_i)$ must also be bounded on the intervals A_0 and A_K by some constants α_0 and α_K respectively. If we let $\alpha = \max_i \{ \frac{N}{N-1} (\alpha_{m,i})^{\frac{1}{N-1}} \}$, then we obtain the desired inequality with $\mu = \alpha$.

Now consider the case $|f'(x_i)| \geq 1$. In this case, the algorithm behaves like gradient descent. And since $|f'(x)| \geq 1$ is satisfied on a finite number of intervals, we can find local Lipschitz constants for f in each of those intervals. DPGD behaves like GD in this case and the local lipchitz property guarantees the descent condition holds in each interval with some constant $\beta_j > 0$.

Finally, we conclude our proof by taking $\mu = \max(\alpha, \beta_1, \ldots, \beta_K)$.

□

B Deep Learning Experiments

Here we provide the hyperparameter tuning spaces for each of our deep learning experiments. Note that we for each task, we fix the same search space for parameters of the same name within different algorithms. For example, all algorithms on the same task use the same search space for the learning rate parameter.

B.1 ResNet-34 on CIFAR-100 - Image Classification

For each algorithm on this task, we tune hyperparameters using 24 trials of random search.

B.2 CharLSTM on Tolstoy's War and Peace - Language Modeling with Recurrent Architecture

For each algorithm on this task, we tune hyperparameters using 100 trials of random search.

Optimizer	Parameters	Tuning Distribution
● GD + Momentum	η	LogUniform(10^{-5}, 1)
	$1-\mu$	LogUniform(10^{-4}, 1)
● RGD + Momentum	η	LogUniform(10^{-5}, 1)
	$1-\mu$	LogUniform(10^{-4}, 1)
	δ	Uniform(0, 30)
● RGD + Momentum Leapfrog	η	LogUniform(10^{-5}, 1)
	$1-\mu$	LogUniform(10^{-4}, 1)
	α	Uniform(0, 1)
	δ	Uniform(0, 30)
● Adam	η	LogUniform(10^{-5}, 1)
	$1-\beta_1$	LogUniform(10^{-3}, 0.6)
	$1-\beta_2$	LogUniform(10^{-3}, 0.4)

Optimizer	Parameters	Tuning Distribution
● GD + Momentum	η	LogUniform(10^{-4}, 1)
	$1-\mu$	LogUniform(10^{-4}, 0.3)
● RGD + Momentum	η	LogUniform(10^{-4}, 1)
	$1-\mu$	LogUniform(10^{-4}, 0.3)
	δ	Uniform(0, 30)
● RGD + Momentum Leapfrog	η	LogUniform(10^{-4}, 1)
	$1-\mu$	LogUniform(10^{-4}, 0.3)
	α	Uniform(0, 1)
	δ	Uniform(0, 30)
● Adam	η	LogUniform(10^{-4}, 1)
	$1-\beta_1$	LogUniform(10^{-3}, 0.3)
	$1-\beta_2$	LogUniform(10^{-3}, 0.2)

B.3 Convolutional VAE on CelebA Faces - Image Generation

For each algorithm on this task, we tune hyperparameters using 24 trials of random search.

OPTIMIZER	PARAMETERS	TUNING DISTRIBUTION
● GD + MOMENTUM	η	LOGUNIFORM(10^{-4}, 10^{-2})
	$1-\mu$	LOGUNIFORM(10^{-4}, 10^{-1})
● RGD + MOMENTUM	η	LOGUNIFORM(10^{-4}, 10^{-2})
	$1-\mu$	LOGUNIFORM(10^{-4}, 10^{-1})
	δ	UNIFORM(0, 30)
● RGD + MOMENTUM LEAPFROG	η	LOGUNIFORM(10^{-4}, 10^{-2})
	$1-\mu$	LOGUNIFORM(10^{-4}, 10^{-1})
	α	UNIFORM(0, 1)
	δ	UNIFORM(0, 30)
● ADAM	η	LOGUNIFORM(10^{-4}, 10^{-2})
	$1-\beta_1$	LOGUNIFORM(10^{-3}, 0.6)
	$1-\beta_2$	LOGUNIFORM(10^{-3}, 0.4)