Born-Infeld (BI) for AI: Energy-Conserving Descent (ECD) for Optimization

G. Bruno De Luca [*1] **Eva Silverstein** [*1]

Abstract

We introduce a novel framework for optimization based on energy-conserving Hamiltonian dynamics in a strongly mixing (chaotic) regime and establish its key properties analytically and numerically. The prototype is a discretization of Born-Infeld dynamics, with a squared relativistic speed limit depending on the objective function. This class of frictionless, energy-conserving optimizers proceeds unobstructed until slowing naturally near the minimal loss, which dominates the phase space volume of the system. Building from studies of chaotic systems such as dynamical billiards, we formulate a specific algorithm with good performance on machine learning and PDE-solving tasks, including generalization. It cannot stop at a high local minimum, an advantage in non-convex loss functions, and proceeds faster than GD+momentum in shallow valleys.

1. Introduction and Summary

Many scientific and technological problems, including machine learning (ML), require optimization, the process of evolving to or near to the minimum of a nontrivial loss function $F(\mathbf{\Theta})$ depending on a high dimensional space of parameters $\mathbf{\Theta}$. The loss function may be highly non-convex and/or contain shallow long valleys. It traditionally works – quite well in many cases – via a form of gradient descent (GD), with many improvements derived over the years involving momentum and adaptive features[1]. A standard example is (stochastic) GD with momentum (Polyak, 1964) which we will denote (S)GDM. In physics terms, these standard algorithms involve a noisy and discretized form of frictional evolution of a particle on a complicated potential energy landscape, without conservation of total kinetic and potential energy E. More precisely, given an objective function $F(\mathbf{\Theta})$, one can define a *potential energy*:

$$V(\mathbf{\Theta}) \equiv F(\mathbf{\Theta}) - \Delta V \,, \tag{1}$$

where ΔV is a (possibly zero) constant shift whose adaptive tuning we will address below. Through this map, optimization can be formulated as (discretized) physical evolution from an initial point $\mathbf{\Theta}_0$, with the goal of reaching sufficiently low values of V. Standard methods correspond to friction-dominated evolution, relying on some form of dissipation of energy to achieve this.

In this work, we show that friction is not necessary: energy-conserving dynamics (ECD) provides a distinctive class of optimization algorithms with some favorable properties, including increased calculability of its behavior, resulting from the E conservation summarized in Table 1. The minimal version contains zero friction, and conserves energy, yet essentially stops moving near vanishing loss. This may sound paradoxical on first glance, but it is straightforward. One example which we focus on in this work is based on relativistic Born-Infeld (BI) dynamics, with the (squared) speed limit[2] c_{rel} depending on $V(\mathbf{\Theta})$ as:

$$c_{rel}^2 = V(\mathbf{\Theta}). \tag{2}$$

As a consequence of this, the evolution essentially stops when $V(\mathbf{\Theta}) \to 0$. As defined in (1), ΔV represents a constant hyper-parameter sometimes needed to shift the objective to $V = 0$; although adaptively tunable, we note that $\Delta V = 0$ in practice for a number of standard problems including ML tasks with achievable vanishing loss objectives and partial differential equation (PDE) solving with F given by the summed squares of the equations. When V vanishes, so does the speed of motion in parameter space. Another example is to take $mass = 1/V$ for ordinary (non-relativistic) particle motion; again even without friction slowing it down, the system ultimately slows as $V \to 0, \text{mass} \to \infty$.

[*]Equal contribution [1]Stanford Institute for Theoretical Physics, Stanford University, Stanford, CA, 94306, USA. Correspondence to: G. Bruno De Luca <gbdeluca@stanford.edu>, Eva Silverstein <evas@stanford.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

[1]See e.g. (Bottou et al., 2018) for a review and (Kingma & Ba, 2015; Reddi et al., 2018) for examples of adaptive optimization.

[2]Relativistic dynamics for optimization was considered previously in the interesting work (França et al., 2020), but there the speed of light was taken to be a constant and friction was introduced to enable the evolution to converge; see also related work in (Lu et al., 2017a). Our approach was motivated by the mechanism (Alishahiha et al., 2004) for early-universe inflationary cosmology, a subject which also involves evolution on a potential landscape.

Table 1. ECD versus frictional optimizers. Energy conservation improves calculational control of optimization and generalization.

ECD	FRICTION ((S)GDM, . . .)
CONSERVES ENERGY E	FRICTION DRAINS E
CANNOT GET STUCK IN HIGH LOCAL MINIMUM	CAN STOP IN HIGH LOCAL MINIMUM
CANNOT OVERSHOOT $V = 0 = \boldsymbol{\nabla} V$	CAN OVERSHOOT $V = 0 = \boldsymbol{\nabla} V$
DEPENDS ON V AND ∇V	DEPENDS ONLY ON $\boldsymbol{\nabla} V$
ON SHALLOW REGION: $\theta \sim e^{-mt/\sqrt{2}}$ (8)	ON SHALLOW REGION: $\theta \sim e^{-m^2 t/f}$ (7)
ANALYTIC PREDICTION FOR DISTRIBUTION	STOCHASTIC INTUITION FOR DISTRIBUTION

In various applications, literally achieving the globally minimal loss may lead to overfitting and failure of generalization. For BI, we find that the speed limit kicks in soon enough to avoid this in small benchmark experiments and our own synthetic PDE solving tests.

Our **contributions** are to introduce ECD, the frictionless energy-conserving class of optimizers with *mixing* dynamics (introduced in §3.1), realized concretely with *BBI* (Algorithm 1), derive its key properties and advantages (Table 1) and confirm them experimentally in synthetic loss functions, PDEs, MNIST and CIFAR.

2. Speed-limited Energy Conserving Frictionless Hamiltonian Dynamics

In physical language, GDM is a discretized version of particle motion with friction on a potential energy function $V(\boldsymbol{\Theta})$. The motion in the potential is governed by Newton's laws of motion, force = mass × acceleration, $-\boldsymbol{\nabla} V - f\dot{\boldsymbol{\Theta}} = m\ddot{\boldsymbol{\Theta}}$ with coefficient of friction f. This is equivalent to the first-order form $\mathbf{p} = m\dot{\boldsymbol{\Theta}}, \dot{\mathbf{p}} = -\mathbf{p}f/m - \boldsymbol{\nabla} V$ in terms of the position $\boldsymbol{\Theta}$ and momentum $\mathbf{p}$, whose appropriate discretization leads to GDM algorithm. Friction leads to violation of energy conservation for the particle (physically, transfer of energy to another sector).

In relativistic particle mechanics, the evolution equations are different, with bounded speed. For squared speed limit V one finds frictionless, energy-conserving continuum evolution equations given in terms of positions θ_i (components of $\boldsymbol{\Theta}$) and momenta $\pi_i \equiv \frac{\dot{\theta}_i}{\sqrt{1-\frac{\dot{\boldsymbol{\Theta}}^2}{V}}}$ (components of the momentum vector $\boldsymbol{\Pi}$):

$$\dot{\theta}_i = \pi_i \frac{V(\boldsymbol{\Theta})}{E}, \quad \dot{\pi}_i = -\frac{\partial_i V}{2}\left(\frac{E}{V} + \frac{V}{E}\right) \tag{3}$$

written in terms of the conserved energy E

$$E = \frac{V}{\sqrt{1-\frac{\dot{\boldsymbol{\Theta}}^2}{V}}} = \sqrt{V(V + \boldsymbol{\Pi}^2)} = \text{constant}. \tag{4}$$

These equations are concisely derived from the appropriate physical *action principle*[3] in Appendix A.1 starting from the action $S = -\int dt V(\boldsymbol{\Theta})\sqrt{1-\frac{\dot{\boldsymbol{\Theta}}^2}{V(\boldsymbol{\Theta})}}$ (Born & Infeld, 1934; Silverstein & Tong, 2004). E conservation ($dE/dt = 0$) in the continuum theory follows from applying (3) to a time derivative of the third expression in (4).

This implies a first-order discrete update rule

$$\pi_i(t+\Delta t) - \pi_i(t) = -\Delta t \frac{\partial_i V(\boldsymbol{\Theta}(t))}{2}\left(\frac{E}{V} + \frac{V}{E}\right) \tag{5}$$

$$\theta_i(t+\Delta t) - \theta_i(t) = \Delta t\, \pi_i(t+\Delta t)\frac{V(\boldsymbol{\Theta}(t))}{E} \tag{6}$$

As we will see more in detail below, noise (minibatches) correspond to a time-dependent sequence of jumps in $V(\boldsymbol{\Theta}, t)$.

This will serve as a starting point for our algorithm, although we will be led to include more features beyond the minimal rule (5)-(6). Specifically, we will rescale $\boldsymbol{\Pi}$ to restore energy conservation once violated by the discrete updates – even in the noisy case of minibatches – and we will include options for randomized bounces at fixed E to avoid stable or quasistable orbits in the motion.

Energy conservation immediately implies

Theorem 2.1. *If $V \neq 0$ and $V \neq E$ then $\dot{\boldsymbol{\Theta}} \neq 0$ in the continuum evolution, and $\boldsymbol{\Theta}(t+\Delta t) \neq \boldsymbol{\Theta}(t)$ for the discrete algorithm.*

Proof. From (4) if $\dot{\boldsymbol{\Theta}} = 0$ and $V \neq 0$ then $E = V$ in the continuum. This extends to the discrete case since the algorithm restores E by rescaling $\boldsymbol{\Pi}$ if numerical errors in energy conservation build up as a result of the discreteness. Specifically, the update rule (5), together with the fact that E is constant, makes it impossible to obtain $\boldsymbol{\Theta}_{t+1} = \boldsymbol{\Theta}_t$ (stalling on parameter space) unless $V = 0$ or $V = E, \boldsymbol{\Pi} = 0$. (The latter only persists to later steps (4) if $\partial V = 0$ with $V = E$ at initialization, a set of measure 0 easily avoided with initial energy δE.) □

Therefore, initialization with $E \geq V \gg V_{\text{objective}}$ ensures the impossibility of getting stuck in a high local minimum with $V \gg V_{\text{objective}}$; illustrated here for BI, this extends to ECD in general.

[3] For a succinct introduction to this general formalism and the role of energy conservation classical physics, see e.g. lectures 6-8 of (Susskind & Hrabovsky, 2013) or (Landau et al., 1976) Chapters 1,2,7.

In various problems, shallow directions in the loss landscape are important (Li et al., 2018; Fort & Scherlis, 2019; Draxler et al., 2018). Here also there is a contrast between ECD (e.g. BI) and GDM as well as their stochastic versions. To get a sense for this, consider a simple shallow direction $V = \frac{1}{2}m^2\theta^2$ (with m^2 the smallest eigenvalue of the Hessian). Gradient descent alone, or friction dominated motion generally, follows the solution

$$\dot{\mathbf{\Theta}} \simeq -\frac{\boldsymbol{\nabla} V}{f} \to \frac{m^2\mathbf{\Theta}}{f}, \quad \mathbf{\Theta} \sim e^{-m^2 t/f} \tag{7}$$

with f the friction coefficient. BI on the other hand satisfies the speed limit

$$|\dot{\mathbf{\Theta}}| \leq \sqrt{V} \to m\theta/\sqrt{2}, \quad \mathbf{\Theta} \sim e^{-mt/\sqrt{2}} \tag{8}$$

For a fixed value of the friction f, as we reduce m so that the potential becomes more shallow, the GD speed decreases quadratically while the BI speed decreases only linearly, thus moving faster toward the objective. We find this distinction survives in the discrete hyper-parameter optimized experiments detailed below.

A related distinction has to do with overshooting $V = 0$. In the continuum, it is impossible for BI to do this since it would violate the speed limit, whereas GDM can overshoot and oscillate (or leave the $V = 0$ basin entirely). The discretized version can overshoot $V = 0$ by a small amount, at which point the algorithm optionally either stops the evolution or adaptively adjusts ΔV if desired (see below).

If initialized at zero velocity ($\dot{\mathbf{\Theta}} = 0$), relativistic effects are negligible to begin with: in that case the Born-Infeld dynamics starts out like a non-relativistic particle exploring the V landscape without friction. Even without friction, the speed limit ensures that it slows dramatically near $V = 0$.

3. Mixing and Chaos: ECD with Dispersing Elements

To avoid ECD (e.g. BI) getting into long-lived orbits high on the loss landscape V, we exploit the analytic power of E conservation and import results from the field of Hamiltonian dynamical systems. This also enables predictions for the late-time distribution of ECD trajectories.

3.1. Mixing and Distributions

The *phase space* of a physical system is the space of accessible momenta $\mathbf{\Pi}$ and positions $\mathbf{\Theta}$. A chaotic system exhibits sensitive dependence on initial conditions, enabling initially closeby trajectories to explore very different regions of the phase space; in its strongest forms most periodic orbits are unstable. For exponentially mixing systems, after a finite timescale t_{mix}, the probability to find a particle – sampled from those in a small initial droplet – in a particular region R of the phase space is given to good approximation by the volume ('measure') of R. [4] This volume is parametrically large as $V \to 0$ for BI. The total phase space volume for an n-dimensional parameter space is given by

$$\text{Vol}(\mathcal{M}) = \int d^n\pi d^n\theta \delta(\sqrt{V(V + \mathbf{\Pi}^2)} - E) \tag{9}$$

Integrating over $\mathbf{\Pi}$ yields eq.(40), concentrated near small V. Near a minimum, V is quadratic, giving (after an orthogonal diagonalization of its Hessian) $V \simeq V_I + \frac{1}{2}\sum_{i=1}^n m_{Ii}^2(\theta_i - \theta_{Ii})^2$. Evaluating this for a basin $|\theta_{Ii} - \theta_i| < 1/m_{Ii}$ at fixed n yields (see §A.3)

$$\text{Vol}(\mathcal{M}_I) \to b_n \left(\frac{2\pi^{n/2}}{\Gamma(n/2)}\right)^2 \frac{E^{n-1}}{\prod_i m_{Ii}} \log(V_I) \quad V_I \to 0, \tag{10}$$

where b_n is a constant. Since $V_I \to 0$ dominates in this formula, mixing with a small enough value of t_{mix} is a sufficient condition for successful optimization in this framework. Since the probability of being in a region of phase space is proportional to its volume, (10) yields an analytic prediction for the distribution of results over multiple low-lying minima which we test experimentally in §4.3. Eq. (10) and the results §4.3 exhibit a preference for low-lying flat minima; on these BBI evolves faster than GD (7)-(8). Other ECD models, e.g. $V \to g(V)$, may enhance this effect. In addition to its use for optimization, the formula enables reverse-engineering an ECD model whose trajectories sample from a desired distribution (De Luca et al., 2022b).

A priori we do not know whether the optimization algorithm combined with the loss function V constitutes a strongly chaotic, exponentially mixing system (although this may be typical), and if so what its value of t_{mix} is. Strong mixing arises in dynamical billiards, see e.g. (Chernov & Markarian, 2006; Szász, 2017). Nearby trajectories bouncing off a billiard ball disperse, leading to sensitive dependence on initial conditions. Inspired by this, we introduce randomized momenta at fixed $\mathbf{\Pi}^2$ in our algorithm (cf. Sec. 3.3).[5]

We stress that the utility of this result depends on the timescale t_{mix}. Consider a system consisting of a box with an arbitrarily tiny hole to a much larger (or even infinite) domain beyond the box. Although the latter has more phase space volume, the time for a random trajectory to hit the hole may be arbitrarily long. So the diverging phase space volume of a region R is not by itself enough to guarantee that the system ends up in R after a small timescale. However, this analogy appears too pessimistic for our case: the gradients introduce forces toward the minimum, and the speed limit on motion is weaker for larger loss V, so the system

[4] A canonical example is the mixing of milk in coffee.

[5] Hamiltonian sampling (Betancourt, 2017; Hanada, 2018) also involves randomized momenta, in general not conserving energy.

moves relatively quickly through such regions, spending more time near $V = 0$. Relatedly, if we endow frictionless momentum (non-relativistic particle motion) with bounces leading to chaos, its version of (10) is *not* dominated by small V (41). This method is based on well developed statistical mechanics intuition and empirical observation, but does not come with a general theoretical guarantee, though it would be interesting to build from the continuum analysis (7)-(8) to seek guarantees in the convex setting though this may be complicated by the nonlinear updates. Typically, the determination of t_{mix} for a complicated system requires experiment, and we will test our intuition that way.

3.2. The Effect of Noise (Minibatches)

Our algorithm will work similarly with or without the use of minibatches, for which the input is a subset of the data updated every N_{batch} steps. With minibatches, the new effect on the dynamics is that the potential $V(\boldsymbol{\Theta}, t)$ becomes a function of time separately from its dependence on $\boldsymbol{\Theta}$. We retain our prescription of explicitly restoring the initial energy E even when minibatches are used; this is a deviation from the motivating physics model, for which explicit time dependence generically leads to a lack of energy conservation. For a loss of the form $\Delta V + V = \Delta V + \sum_{\text{batches}} F_{\text{batch}}$ with $F_{\text{batch}} \geq 0 \ \ \forall$ batches, Theorem 2.1 continues to apply.

We will explore aspects of the behavior of BI with minibatches at greater length in §A.4, both from this deterministic (but time-dependent) perspective, and from the perspective of appropriate averaged stochastic dynamics; the speed limit $\dot{\boldsymbol{\Theta}}^2 < V$ constrains the variance of $\dot{\boldsymbol{\Theta}}$.

3.3. The BBI Algorithm

We now combine the ingredients introduced so far and describe our detailed optimization algorithm. As explained in Sec. 2, the starting point is given by the the continuum equations (3). The first step is their discretization, which we perform using a first order *symplectic method*.[6] This ensures that at this stage volume density in phase space is preserved (cf. (França et al., 2020), Sec. 3), and results in the update rules

$$\boldsymbol{\Pi}_{k+1} = \boldsymbol{\Pi}_k - \frac{1}{2}\boldsymbol{\nabla} V_k \Delta t \left(\frac{E}{V_k} + \frac{V_k}{E}\right) \tag{11}$$

$$\boldsymbol{\Theta}_{k+1} = \boldsymbol{\Theta}_k + \boldsymbol{\Pi}_{k+1} \Delta t \frac{V_k}{E} \ , \tag{12}$$

where the subscript k refers to the k-th iterative step, and $V_k = V(\boldsymbol{\Theta}_k, t_k)$ (with the explicit t-dependence applicable in the case with minibatches discussed in §3.2 and §A.4). The energy E is a constant defined as

$$E \equiv V_0 + \delta E \ , \tag{13}$$

where V_0 is the initial value of the (shifted) objective function (2) at the beginning of the evolution and $\delta E \geq 0$ is an optional extra hyperparameter. Its primary role is to add an initial energy to overcome possible energy barriers higher than the initial value of the potential, which is useful in highly non-convex problems, such as the Ackley function discussed in Sec. 4.1. It can also introduce more chaos in the system to help distribute the results.

As shown in Theorem 2.1, the dynamics just described stops changing $\boldsymbol{\Theta}$ appreciably only when $V = F - \Delta V \to 0$ where ΔV is the hyperparameter defined in (2). This acts both as threshold for the required accuracy of the solution and as a shift in case the expected optimum $F(\boldsymbol{\Theta}) \neq 0$.

In fact an adaptive tuning of ΔV is possible, added as an option to the altorithm and tested in computational chemistry experiments (De Luca et al., 2022c). Given a too-high initial guess, the loss extends to $V = F - \Delta V < 0$ and the trajectory will jump to a small negative value $V < 0$ due to the discreteness. Conditioned on this, ΔV may be lowered; this iteratively tunes it.

The vector $\boldsymbol{\Theta}$ is initialized by choosing an initial point in parameter space (e.g. a standard initialization of a Neural Network (NN)). The vector $\boldsymbol{\Pi}$ can be initialized in various ways consistent with energy conservation. A natural choice is to initialize it in the direction of (minus) the initial gradient,

$$\boldsymbol{\Pi}_0 \equiv -\frac{\boldsymbol{\nabla} V(\boldsymbol{\Theta}_0)}{|\boldsymbol{\nabla} V(\boldsymbol{\Theta}_0|} \sqrt{\frac{E^2}{V_0} - V_0}, \tag{14}$$

with norm satisfying (4). Another possibility is to initialize it in a random direction. Notice that if $\delta E = 0$, $\boldsymbol{\Pi}_0$ is always initialized to zero, as is common in optimization.

The optimizer described so far implements the BI dynamics, but does not yet contain the extra features encouraging chaotic mixing as discussed in §3.1. This can be achieved by introducing random bounces, which conserve energy by keeping $\boldsymbol{\Pi}^2$ the same in the process. Such bounces can be easily implemented by generating a new random $\boldsymbol{\Pi}$ with the same norm as the original $\boldsymbol{\Pi}$. We implement these in two ways: (i) N_b fixed bounces separated by T_0 timesteps, and (ii) A variable number of progress-dependent bounces that are performed if for T_1 timesteps there is no progress in the search of the minimum, e.g. if it fails to reach a value of V smaller than all those previously seen. We will refer to this Bouncing BI dynamics as *BBI*.

As a final ingredient, we include a rescaling of $\boldsymbol{\Pi}$ that restores possible energy conservation violations due to numer-

[6] *Symplectic methods* are integration methods that preserve the geometric structure of Hamiltonian dynamics. They are relevant for our discussion since this also ensures that phase space volumes are preserved. See e.g. (Hairer et al., 2006) for a review.

ical and discretization effects and minibatch transitions.[7] This is performed by computing the current value of $\mathbf{\Pi}_k^2$ at each step and comparing it with $\sqrt{\frac{E^2}{V_k} - V_k}$, the value it should have at step k. If there is a discrepancy, we homogeneously rescale all the components of $\mathbf{\Pi}_k$ to restore $\mathbf{\Pi}_k^2$ to the appropriate value. The rescaling is not performed if the quantity in the square root happens to be negative, something that could happen very early in the evolution due to small numerical errors. This mechanism, not entirely faithful to the continuum dynamics, introduces small changes in the phase space measure. We expect this effect to be small and not appreciably affect the analytic estimates, confirming this empirically in quantitative experiments in Sec. 4.3. Algorithm 1 summarizes BBI.

4. Experiments

In this section, we empirically confirm the theoretically predicted behavior of BBI as an example of (S)ECD, and compare it to (S)GDM, studying both the noise-free and noisy (stochastic) cases. We do not systematically compare to adaptive optimizers such as Adam (Kingma & Ba, 2015), leaving such features in BBI to future work. In §4.1-4.3 we analyze synthetic functions useful for testing optimization algorithms, illustrating the theory derived in previous sections and testing BBI against GDM. In §4.4-4.5 we present richer, high-dimensional experiments in PDE solving and ML (MNIST (Lecun et al., 1998) and CIFAR (Krizhevsky et al., 2009)). These include minibatches, enabling us to confirm the robustness of BBI in that setting.

We ran the synthetic experiments and MNIST on standard laptop CPUs, while for CIFAR and the PDEs we used two GPUs. More details, including the full source code and sample results, can be found at https://github.com/gbdl/BBI.

4.1. Highly Non-Convex Landscapes.

In this section we consider highly non-convex functions, focusing on the hard-to-optimize Ackley function (Ackley, 2012), plotted in Fig. 1 (left) and defined as

$$F(\theta_1,\theta_2) \equiv -20\exp\left[-0.2\sqrt{0.5\,(\theta_1^2+\theta_2^2)}\right] + \\ -\exp\left[0.5\,(\cos 2\pi\theta_1 + \cos 2\pi\theta_2)\right] + e + 20\,. \quad (15)$$

We test the BBI optimizer on this function, to check its ability to escape the many local minima and to compare it with friction-based methods. A typical evolution is displayed in Fig. 2. Starting from a fixed random point, we observe:

- If δE is not large enough to escape the initial local

[7] Recall from §2 that the continuum dynamics conserves energy exactly in the noise-free case.

Algorithm 1 *BBI*. ε_1 and ε_2 are numerical constants ensuring stability (good default values are $\varepsilon_1 = 10^{-10}$, $\varepsilon_2 = 10^{-40}$). The hyperparameter ΔV can be adaptively tuned as explained in the main text. See also the code for a preliminary working implementation.

Require: Δt: Stepsize
Require: ΔV: Shift hyperparameter.
Require: δE: Extra initial energy.
Require: T_0, T_1, N_b: Bouncing thresholds and # of fixed bounces.
Require: $F(\mathbf{\Theta})$: Function to minimize.
Require: $\mathbf{\Theta}_0$: Initial parameter vector.
$\{c_0, c_1, n_b\} \leftarrow \{0,0,0\}$ (Initialize counters for bounces)
$V \leftarrow F(\mathbf{\Theta}_0) - \Delta V$ (Initialize V)
$E \leftarrow V + \delta E$ (Initialize energy)
$\mathbf{\Pi}_0 \leftarrow -\frac{\boldsymbol{\nabla} F(\mathbf{\Theta}_0)}{|\boldsymbol{\nabla} F(\mathbf{\Theta}_0)|}\sqrt{\frac{E^2}{V} - V}$ (Initialize momenta)
$t \leftarrow 0$ (Initialize timestep)
while $V > \varepsilon_2$ **do**
  if $c_0 \neq T_0$ **and** $c_1 \neq T_1$ **then**
    $t \leftarrow t+1$
    $\pi_C^2 \leftarrow V(\frac{E^2}{V^2} - 1)$ (Correct value of $\mathbf{\Pi}^2$)
    if $|\mathbf{\Pi}_{t-1}^2 - \pi_C^2| < \varepsilon_1$ **or** $\pi_C^2 < 0$ **then**
      $\alpha \leftarrow 1$
    else
      $\alpha \leftarrow \sqrt{\frac{\pi_C^2}{\mathbf{\Pi}_{t-1}^2}}$ (Factor to restore energy cons.)
    end if
    $\mathbf{\Pi}_t \leftarrow \alpha \mathbf{\Pi}_{t-1}$ (Restore energy conservation)
    $\mathbf{\Pi}_t \leftarrow \mathbf{\Pi}_t - \frac{1}{2}\Delta t(\frac{V}{E} + \frac{E}{V})\boldsymbol{\nabla} F(\mathbf{\Theta}_{t-1})$
    $\mathbf{\Theta}_t \leftarrow \mathbf{\Theta}_{t-1} + \Delta t \frac{V}{E}\mathbf{\Pi}_t$
    $c_0 \leftarrow c_0 + 1$
    $c_1 \leftarrow c_1 + 1$
    if *new minimum* **then**
      $c_1 \leftarrow 0$
    end if
    $V \leftarrow F(\mathbf{\Theta}_t) - \Delta V$
  else
    $\mathbf{\Pi}_N \leftarrow$ *random vector* (Generate random vector)
    $\mathbf{\Pi}_t \leftarrow \mathbf{\Pi}_N \sqrt{\frac{\mathbf{\Pi}_t^2}{\mathbf{\Pi}_N^2}}$ (Bounce momenta cons. $\mathbf{\Pi}^2$)
    if $c_0 = T_0$ **then**
      $n_b \leftarrow n_b + 1$
      if $n_b < N_b$ **then**
        $c_0 \leftarrow 0$
      else
        $c_0 \leftarrow c_0 + 1$
      end if
    end if
    $c_1 \leftarrow 0$
  end if
end while

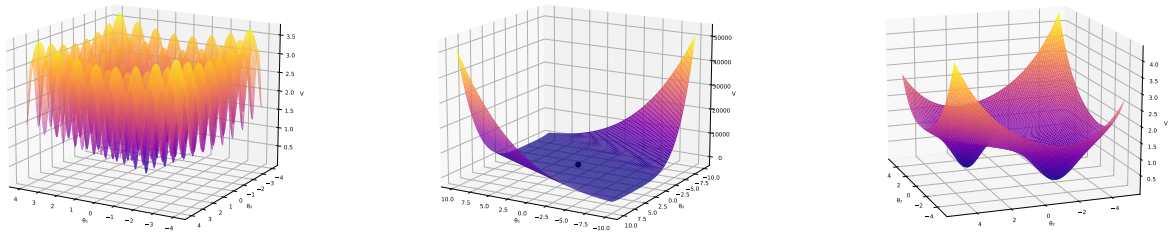

Figure 1. Synthetic benchmark functions (depicted in 2 dimensions). A highly non-convex function (left), a function with shallow valleys (center) and a simple multi-basin function (right).

minimum, the particle keeps oscillating indefinitely. This is consistent with conservation of energy.

- When δE is above a certain threshold, the particle escapes the first local minimum and keeps moving, eventually discovering the global minimum at $\theta_i = 0$.

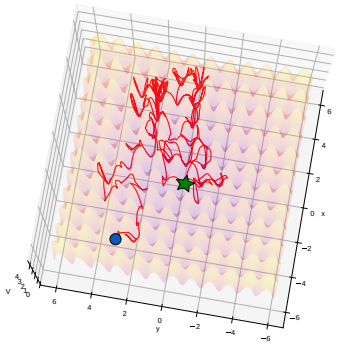

Figure 2. A typical BBI trajectory (starting from the circle) explores the rich landscape, stopping only at the global minimum (star).

This is to be contrasted with friction based optimizers, such as GDM. In that case, we observe three regimes: (i) with a small stepsize, the evolution is not able to escape from the first local minimum it finds, (ii) with higher stepsizes, the optimizer jumps out of the first local minimum and keeps exploring the landscape erratically. Doing so it might be able to get somewhat close the the global minimum, but the evolution is unstable and does not converge to that point, (iii) with even higher learning rates the evolution is divergent. This behavior is consistent with the analysis in (Lewkowycz et al., 2020), where the second phase is called *catapult phase*. We will adopt the same name[8].

[8](Lewkowycz et al., 2020) found that on certain problems with flat minima the catapult phase is best for (S)GD. See Sec. 4.2 for a function with shallow regions.

To confirm this behavior in general we performed a search in the hyperparameter space by using hyperopt with its Tree Of Parzen Estimators algorithm (Bergstra et al., 2013). To do so, we picked an initial point, and used hyperopt to find the best hyperparameters for the various optimizers. For each optimizer we ran 500 trials, evolving for 200 steps each, using the lowest point found during this evolution as the measure of success. For GDM we hyperoptimized both the learning rate $\eta \in [10^{-4}, .5]$ and the value of the momentum $\mu \in [.0, 1.0]$.[9] For BBI, we fixed a default value for the chaos-inducing hyperparameters ($T_0 = 20, N_b = 4, T_1 = 100$), set $\delta E = 2$ to overcome the initial barrier, and used hyperopt only on the step size. The results confirm the observations above. Indeed, for GDM to escape the initial minimum, hyperopt must increase the step-size, which then results in an erratic unstable evolution. Since hyperopt tries to avoid divergences, it will automatically select the catapult phase to escape the initial minimum and get to a lower loss. To confirm this interpretation, we also ran hyperopt again on GDM but with a more stringent upper bound on the allowed value of step-size, finding that with this constraint it is not able to escape from the initial local minimum.

After this phase of estimation of the best hyperparameters, we ran the evolution again for a longer time. GDM in the catapult regime keeps moving erratically, while BBI explores the rich landscape eventually finding the global minimum. Fig. 3 summarizes these results.

Finally, we also checked that these results are robust against changes of the initial point. To confirm this, we evolved from 100 randomly selected points for $(x, y) \in [-4, 4]^2$ using the hyperparameters estimated above. For each of the points we ran the evolution 5 times, to take advantage of the chaotic (and non-deterministic) features of BBI, but not during the estimation phase. Doing this, we obtained that for 90/100 random initial points, BBI achieved $V < 5 \times 10^{-4}$

[9]We also ran hyperopt a second time on the pair $\{\eta, \gamma = -\log \mu\}$, not finding an appreciable difference (in other experiments this improved GD e.g. compared to (França et al., 2020)). For (S)GDM and its parameters we are using the default implementation in Pytorch.

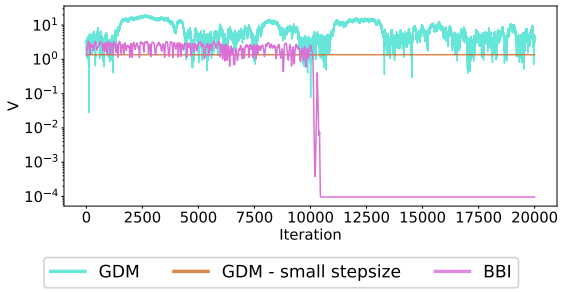

Figure 3. Comparison of the optimizers on the highly non-convex Ackley function (15). The plot shows a typical run with the best hyperparameters as estimated with hyperopt. GDM is either stuck on the initial local minimum or it catapults itself out with a high step size, which however forbids it to converge to the global optimum. BBI explores the rich landscape eventually finding the global minimum and converging to a prescribed accuracy specified by ΔV (here 10^{-4}).

within 3×10^4 iterations, compared to 2/100 of the times where GDM gets close to the minimum by chance during its erratic evolution.

4.2. Shallow Valleys

In this section, we analyze the case where the global minimum lies in an almost-flat valley, making efficient convergence hard. For concreteness, we focus on the Zakharov function ((Jamil & Yang, 2013), Function 173), plotted in 2 dimensions in Fig. 1 (center), with n-dimensional definition

$$F(\mathbf{\Theta}) \equiv \sum_{i=1}^{n} \theta_i^2 + \left(\frac{1}{2}\sum_{i=1}^{n} i\theta_i\right)^2 + \left(\frac{1}{2}\sum_{i=1}^{n} i\theta_i\right)^4 . \quad (16)$$

It has no local minima, but the global minimum at $\mathbf{\Theta} = 0$ lies in a nearly flat valley, slowing optimization. We take $n = 10$, and compare the performance of BBI and GDM with the same methodology described in Sec. 4.1. That is, we start from a fixed reference point, $(-1, ..., -1)$, and use hyperopt to estimate the best hyperparameters by evolving 500 times for 2500 iterations. For GDM we search on the stepsize twice ($\eta \in [10^{-10}, .5]$ and $\eta \in [10^{-10}, 10^{-5}]$) and on the momentum $\mu \in [.0, 1.0]$, while for BBI we set high thresholds to turn off bounces, set $\delta E = 0$, and search only on the step-size $\Delta t \in [10^{-6}, 10^{-2}]$. With the best hyperparameters thus determined, we ran the evolution for 10^4 iterations, obtaining the results in Fig. 4. We also selected random initialization points and ran multiple evolutions with the previously estimated best hyperparameters, confirming that BBI consistently performs better than GDM.

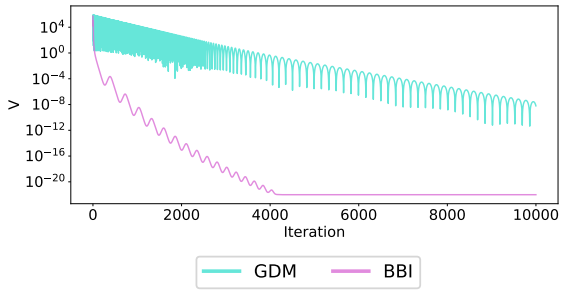

Figure 4. Performance comparison on a ten-dimensional Zakharov function with the best estimated parameters. BBI proceeds faster and stops at the desired accuracy specified by ΔV, here 10^{-22}.

4.3. Two-Dimensional Multi-Basin Function

In this section we study the optimization in a simple landscape with two global minima whose basins of attraction have different shapes. In this landscape, BI alone (without bounces) does not exhibit chaotic mixing in our experiments. We will confirm empirically that the chaotic billiards-inspired bouncing prescription in BBI is able to find multiple solutions of global optimization problems which matches the predictions from mixing dynamics. Explicitly, we consider the function

$$\begin{aligned} F(\mathbf{\Theta}) \equiv & -\exp\left(-0.4|\mathbf{\Theta} - \mathbf{c_1}|^2\right) + \\ & - (1-\epsilon)\exp\left(-0.8|\mathbf{\Theta} - \mathbf{c_2}|^2\right) + \\ & + 10^{-3}|\mathbf{\Theta} - \mathbf{c_1}|^2|\mathbf{\Theta} - \mathbf{c_2}|^2 + 1 , \end{aligned} \quad (17)$$

where $\epsilon \ll 1$ can be tuned to ensure the two minima are numerically at the same height, and $\mathbf{c_1}$ and $\mathbf{c_2}$ are the locations of the two global minima. We will use $\epsilon \sim 10^{-6}$, $\mathbf{c_1} = (-2, -2)$, $\mathbf{c_2} = (2, 2)$. This function has two global minima for which $V = 0$ (see Fig. 1 (right)).

The shapes of the basins directly enter in the volume formula (10) through the eigenvalues of the Hessian $H_{ij} \equiv \partial^2_{ij} V$, $i, j = \{1, 2\}$. For the potential (17) this gives the ratio of the volumes of the two basins

$$\mathrm{Vol}(\mathcal{M}_1)/\mathrm{Vol}(\mathcal{M}_2) = \sim 1.93 , \quad (18)$$

assuming the energy is held fixed. In a perfectly mixing system this ratio is also the ratio of convergence to the two basins. To test how close the bouncing prescription in the BBI algorithm brings the potential (17) to a mixing system, we performed multiple evolutions and checked the ratio of times the particles get in one of the two basins. With 10^3 evolutions, the partial ratios already asymptote to a value close to the theoretically predicted value (18). For the example in Fig. 6, the agreement is within $\sim 11\%$.

This shows that the addition of bounces introduces chaos even in a simple non-mixing potential for pure BI. This

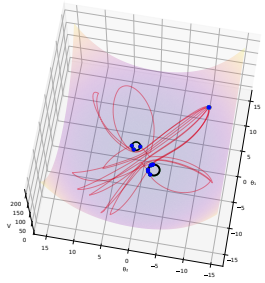

Figure 5. Ten trajectories starting from the same initial point, bouncing at the beginning. The black circles denote boundaries of the regions of basins. This shows that an initial random bounce and a small number of other bounces along the evolution are able to distribute the trajectories in the different basins.

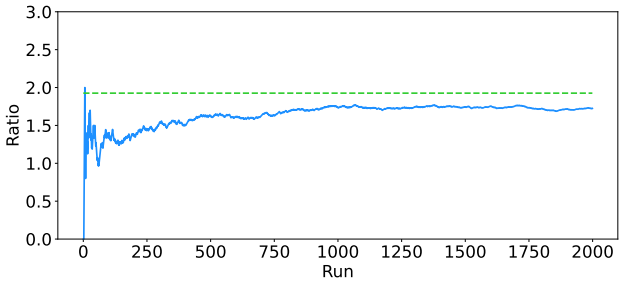

Figure 6. Partial ratios of convergence to the different basins on multiple runs (blue). All the runs start from the same initial point and are performed with $N_b = 1, \Delta t = 10^{-2}, T_1 = 750, T_2 = 20, \delta E = 0, \Delta V = 10^{-3}$. The asymptotic ratio is within 11% of the expected ratio for a mixing system (Eq. (18), green).

implies that BBI can find multiple solutions of optimization problems, a result we will reproduce in §4.4 in a PDE-solving example. The analytic estimate for the ratio of trajectories landing on different minima (10) holds up in this simple experiment. The formula (10) assumes effective mixing on a relevant timescale and requires knowledge of the geometry of the basins near each $V = V_I$. Since these are not a priori known in many problems, it might be hard to check for some situations. In any case, the present experiment shows that the bounces can produce approximately mixing systems such that the optimization converges to the different solutions with good control. Here we make no comparison to GD-like optimizers, which lack a prediction of the ratio of convergence to different basins.

4.4. PDE Solving Examples

In this section, we start the investigation of larger optimization problems, solving Partial Differential Equations (PDEs) with NNs. PDEs are ubiquitous in science; developing efficient numerical solvers is of paramount importance. Many methods have been developed to attack this problem, including recent exploration of Machine Learning methods. (A partial list includes (Lagaris et al., 1998; Weinan & Yu, 2017; Sirignano & Spiliopoulos, 2018; Raissi et al., 2019; Bar-Sinai et al., 2019; Piscopo et al., 2019; Geist et al., 2021; Zang et al., 2020; Lu et al., 2020) and (Beck et al., 2021; Lu et al., 2022).) These promise more versatility, potentially avoiding the *curse of dimensionality*, the exponential increase in complexity with the number of dimensions.

The most common approach is to employ a NN as an ansatz for the solution of the PDE, using the squared PDE itself (together with its boundary conditions (BC)) as a loss function, schematically $V(\boldsymbol{\Theta}) = F(\boldsymbol{\Theta})$ is given by:

$$\sum_{x\in\mathcal{D}} \text{PDE}[\mathcal{N}(x;\boldsymbol{\Theta})]^2 + \gamma \sum_{x\in\partial\mathcal{D}} \text{BC}[\mathcal{N}(x;\boldsymbol{\Theta})]^2 + \text{R}(\boldsymbol{\Theta}) \tag{19}$$

where $\mathcal{N}$ is a neural network, γ is a fixed coefficient, R is a possible regularization term and $\mathcal{D}$ and $\partial\mathcal{D}$ are sets of points randomly sampled from the domain of the PDE and its boundary. We note that here the ΔV parameter (2) is not needed and is set to 0. The price paid by these methods is the absence of guarantees about the convergence and accuracy. Indeed, even though NNs can represent large classes of functions with good accuracy (Cybenko, 1989; Hornik et al., 1989; Lu et al., 2017b; Liang & Srikant, 2017) , the fact that the training converges to those is not guaranteed (see e.g. (Adcock & Dexter, 2021) for a recent analysis of convergence of Neural Networks to known functions), and the optimizer is important.

In this regard, the BBI properties of not stopping until the best approximation to the solution is found and of predictably exploring different regions of the parameter space to capture multiple solutions appear very promising. We have verified this by constructing a hard PDE problem, with two known distinct solutions, finding that BBI with a residual NN ansatz in (19) successfully converges to both solutions with good accuracy consistently across hundreds of experiments. See App. B for details on the PDE problem.

The randomly sampled points x can be changed during the evolution, similarly to minibatches in ML problems, giving rise to a time-dependent potential $V(\boldsymbol{\Theta}, t)$ as described in §3.2. Doing this every epoch =1000 iterations, with $\Delta V = 0, N_b = 10, \Delta t = 5 \times 10^{-6}, \delta E = 2 \times 10^6, T_0 = 1000$, and $T_1 = 5000$ for a small sample of 12 1500-epoch runs yields consistent solutions of the PDE, dominated by one solution (finding the other solution 1 out of 12 times). Although small statistics, this illustrates BBI's robustness against batch-induced noise.

Indeed in BBI the effects of noise are limited by the speed limit on the variance of $\dot{\boldsymbol{\Theta}}$ (see §3.2 and §A.4). We find multiple PDE solutions due to the bouncing (without noise)

Table 2. Test set accuracy: Mean , Median.

DATA SET	SGD	BBI
MNIST	99.166 , 98.160	99.177 , 99.190
CIFAR-10	92.628 , 92.655	92.434 , 92.435

from the same initial point. It would be interesting to quantify this analysis (c.f. §4.3) for these larger problems.

4.5. Small-Scale ML Benchmarks

Next we consider MNIST (Lecun et al., 1998) and CIFAR-10 (Krizhevsky et al., 2009) as small ML benchmarks on which to test BBI, giving another check of whether the prescription of enforcing energy conservation described in Sec. 3.3 works well with minibatches (Table 2).

For MNIST we used a simple CNN having two convolutional layers with maxpool in between and a final dense layer. We took batches of size 50 and performed a grid search on the hyperpameters running for 3 epochs. For SGD we searched on the learning rate $\eta \in \{.001, .01, .05, .1, .2, .3\}$ and on the momentum $\mu \in \{.85, .9, .95, .99, .999\}$, while for BBI we scanned over the same learning rates $\Delta t \in \{.001, .01, .05, .1, .2, .3\}$ and on $\delta E \in \{.0, .1, .5, 1.0, 2.0\}$, keeping fixed $\Delta V = 10^{-6}, T_0 = 100, N_b = 5, T_1 = 1000$. Choosing the best parameters according to their test accuracy, we performed 60 evolutions for 50 epochs, recording mean and median of the accuracy on the test set.

For CIFAR-10, we used a ResNet-18 (He et al., 2016) and employed hyperopt to estimate the best hyperperameters by performing 50 trials for 3 epochs each. For SGD we searched both on $\eta \in [0.001, 0.2]$ and $\mu \in [.8, 1.0]$ while for BBI we fixed the chaos-inducing parameters ($N_b = 100, T_0 = 100, T_1 = 200$) and $\Delta V = \delta E = 0$ and only searched on $\Delta t \in [.001, .2]$. After the estimation phase, we evolved 16 times for 150 epochs with the best hyperparameters and computed mean and median. Both benchmarks use a standard Cross Entropy loss. Given the small statistics, the distributions of results are not Gaussian and we only report mean and median.

We stress that the experiments in this section are not meant as a thorough comparison, but only as a first check that the BBI algorithm performs well on larger scale problems and is robust to the presence of minibatch. This also shows that overfitting does not appear to be a problem – as anticipated given from the speed limit strongly slowing evolution before literally $V = 0$ – since the accuracy results on the test set are in line with SGDM. In progress are extensions to larger ML problems, analyzing ECD dynamics (10) in relation to properties of the loss $F(\boldsymbol{\Theta})$ (De Luca et al., 2022a).

5. Discussion

We have established that maintaining energy conservation in the descent of the loss function V is compatible with successful optimization, leading to certain analytic handles on the process that are unavailable for frictional methods (SGDM etc.). Small-scale and benchmark numerical experiments bear out the theoretical predictions and intuitions, with some favorable properties and results compared to the standard frictional optimization. There is much more to explore particularly with larger-scale applications in order to determine the full impact of these novel features.

An important direction is to optimize among ECD optimizers, with the guidance of (10)(40). Combining ECD with adaptive stepsizes as in e.g. (Kingma & Ba, 2015) or with weight averaging (Izmailov et al., 2018) – distributed predictably as in (40)(10) – may be worthwhile. It is intriguing that the stochasticity of SGD, which introduces hard-to-characterize noise, is largely replaced in BBI by the bounces; the relativistic speed limit constrains the variance $\langle \dot{\boldsymbol{\Theta}}^2 \rangle$ entering into minibatch-induced noise (§A.4). Since the randomness induced by bouncings is intrinsic (not tied to the batches), we expect that bouncing ECD algorithms are robust against adversarial attacks based on batch ordering such as (Shumailov et al., 2021). It will also be interesting to assess the way that ECD's distinctive dynamics affects representation learning, e.g. adapting to ECD the frameworks developed in (Roberts et al., 2021) and (Yang et al., 2022) for initialization and hyperparameter scalings with network size.

Acknowledgements

We are grateful to the reviewers for their stimulating comments and questions. We would like to thank Dan Roberts for many useful discussions and for sharing with E.S. his work with Josh Batson and Yoni Kahn connecting inflationary theory and optimization, stimulating this work. We are very grateful to George Panagopoulos for early collaboration and numerous contributions, along with Thomas Bachlechner for fruitful intermediate collaboration. We thank many others for useful discussions including Guy Gur-Ari, Miranda Cheng, Ethan Dyer, Alice Gatti, Daniel Kunin, Aitor Lewkowycz, Luisa Lucie-Smith, Hiranya Peiris, Andrew Pontzen, Uros Seljak, Vasu Shyam, Kendrick Smith, Scott Tremaine, Sho Yaida, Zhiyong Zhang and participants of the 2021 Modern Inflationary Cosmology and String-Data workshops. Our research is supported in part by the Simons Foundation Investigator and Modern Inflationary Cosmology programs, and the National Science Foundation under grant number PHY-1720397. Some computing was performed on the Sherlock cluster. We thank Stanford University and the Stanford Research Computing Center for support and computational resources.

References

Ackley, D. *A Connectionist Machine for Genetic Hill-climbing*. The Springer International Series in Engineering and Computer Science. Springer US, 2012. ISBN 9781461319979.

Adcock, B. and Dexter, N. The gap between theory and practice in function approximation with deep neural networks. *SIAM Journal on Mathematics of Data Science*, 3 (2):624–655, 2021. doi: 10.1137/20M131309X.

Alishahiha, M., Silverstein, E., and Tong, D. DBI in the sky. *Phys. Rev. D*, 70:123505, 2004. doi: 10.1103/PhysRevD.70.123505.

Armendariz-Picon, C., Damour, T., and Mukhanov, V. F. k - inflation. *Phys. Lett. B*, 458:209–218, 1999. doi: 10.1016/S0370-2693(99)00603-6.

Bar-Sinai, Y., Hoyer, S., Hickey, J., and Brenner, M. P. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31):15344–15349, Jul 2019. ISSN 1091-6490. doi: 10.1073/pnas.1814058116. URL http://dx.doi.org/10.1073/pnas.1814058116.

Beck, C., Becker, S., Grohs, P., Jaafari, N., and Jentzen, A. Solving the kolmogorov pde by means of deep learning. *J. Sci. Comput.*, 88:73, 2021.

Bergstra, J., Yamins, D., and Cox, D. D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pp. I–115–I–123. JMLR.org, 2013.

Betancourt, M. A Conceptual Introduction to Hamiltonian Monte Carlo. *arXiv e-prints*, art. arXiv:1701.02434, January 2017.

Born, M. and Infeld, L. Foundations of the new field theory. *Proceedings of the Royal Society A*, 144:425–451, 1934.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018. doi: 10.1137/16M1080173.

Chen, X., Huang, M.-x., Kachru, S., and Shiu, G. Observational signatures and non-Gaussianities of general single field inflation. *JCAP*, 01:002, 2007. doi: 10.1088/1475-7516/2007/01/002.

Chernov, N. and Markarian, R. Chaotic billiards. In *Mathematical Surveys and Monographs, 127. American Mathematical Society*, 2006.

Cheung, C., Creminelli, P., Fitzpatrick, A. L., Kaplan, J., and Senatore, L. The Effective Field Theory of Inflation. *JHEP*, 03:014, 2008. doi: 10.1088/1126-6708/2008/03/014.

Cybenko, G. V. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

De Luca, G. B., Kunin, D., and Silverstein, E. Testing energy conserving descent on machine learning benchmarks (work in progress), 2022a.

De Luca, G. B., Robnik, J., Seljak, U., and Silverstein, E. Energy conserving descent for sampling (work in progress), 2022b.

De Luca, G. B., Silverstein, E., and Zhang, Z. Energy conserving descent for computational chemistry (work in progress), 2022c.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pp. 1309–1318. PMLR, 2018.

Fort, S. and Scherlis, A. The goldilocks zone: Towards better understanding of neural network loss landscapes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3574–3581, Jul. 2019. doi: 10.1609/aaai.v33i01.33013574. URL https://ojs.aaai.org/index.php/AAAI/article/view/4237.

França, G., Sulam, J., Robinson, D. P., and Vidal, R. Conformal symplectic and relativistic optimization. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124008, Dec 2020. ISSN 1742-5468. doi: 10.1088/1742-5468/abcaee. URL http://dx.doi.org/10.1088/1742-5468/abcaee.

Geist, M., Petersen, P., Raslan, M., Schneider, R., and Kutyniok, G. Numerical solution of the parametric diffusion equation by deep neural networks. *Journal of Scientific Computing*, 88(1):1–37, 2021.

Hairer, E., Hochbruck, M., Iserles, A., and Lubich, C. Geometric numerical integration. *Oberwolfach Reports*, 3(1):805–882, 2006.

Hanada, M. Markov chain monte carlo for dummies, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hornik, K., Stinchcombe, M. B., and White, H. L. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. Averaging weights leads to wider optima and better generalization. In Silva, R., Globerson, A., and Globerson, A. (eds.), *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, 34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018, pp. 876–885. Association For Uncertainty in Artificial Intelligence (AUAI), 2018.

Jamil, M. and Yang, X.-S. A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150–194, 2013.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kunin, D., Sagastuy-Breña, J., Gillespie, L., Margalit, E., Tanaka, H., Ganguli, S., and Yamins, D. L. K. Rethinking the limiting dynamics of SGD: modified loss, phase space oscillations, and anomalous diffusion. *CoRR*, abs/2107.09133, 2021. URL https://arxiv.org/abs/2107.09133.

Lagaris, I., Likas, A., and Fotiadis, D. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9 (5):987–1000, 1998. doi: 10.1109/72.712178.

Landau, L., Lifshitz, E., Sykes, J., and Bell, J. *Mechanics: Volume 1*. Course of theoretical physics. Elsevier Science, 1976. ISBN 9780750628969.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Lewkowycz, A., Bahri, Y., Dyer, E., Sohl-Dickstein, J., and Gur-Ari, G. The large learning rate phase of deep learning: the catapult mechanism. *CoRR*, abs/2003.02218, 2020. URL https://arxiv.org/abs/2003.02218.

Li, C., Farkhoor, H., Liu, R., and Yosinski, J. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018.

Liang, S. and Srikant, R. Why deep neural networks for function approximation? In *ICLR*, 2017.

Lu, L., Meng, X., Mao, Z., and Karniadakis, G. E. Deepxde: A deep learning library for solving differential equations. *SIAM Rev.*, 63:208–228, 2020.

Lu, X., Perrone, V., Hasenclever, L., Teh, Y. W., and Vollmer, S. Relativistic monte carlo. In *Artificial Intelligence and Statistics*, pp. 1236–1245. PMLR, 2017a.

Lu, Y., Chen, H., Lu, J., Ying, L., and Blanchet, J. Machine learning for elliptic PDEs: Fast rate generalization bound, neural scaling law and minimax optimality. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=mhYUBYNoGz.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The expressive power of neural networks: A view from the width. In *NIPS*, 2017b.

Mathis, D., Mousatov, A., Panagopoulos, G., and Silverstein, E. A new branch of inflationary speed limits. *JHEP*, 10:199, 2021. doi: 10.1007/JHEP10(2021)199.

Muehlebach, M. and Jordan, M. I. Optimization with momentum: Dynamical, control-theoretic, and symplectic perspectives. *Journal of Machine Learning Research*, 22 (73):1–50, 2021.

Piscopo, M. L., Spannowsky, M., and Waite, P. Solving differential equations with neural networks: Applications to the calculation of cosmological phase transitions. *Phys. Rev. D*, 100(1):016002, 2019. doi: 10.1103/PhysRevD.100.016002.

Polyak, B. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. ISSN 0041-5553. doi: https://doi.org/10.1016/0041-5553(64)90137-5.

Raissi, M., Perdikaris, P., and Karniadakis, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

Reddi, S. J., Kale, S., and Kumar, S. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ryQu7f-RZ.

Roberts, D. A., Yaida, S., and Hanin, B. The Principles of Deep Learning Theory. 6 2021. doi: 10.1017/9781009023405.

Shumailov, I., Shumaylov, Z., Kazhdan, D., Zhao, Y., Papernot, N., Erdogdu, M. A., and Anderson, R. J. Manipulating sgd with data ordering attacks. *Advances in Neural Information Processing Systems*, 34:18021–18032, 2021.

Silverstein, E. and Tong, D. Scalar speed limits and cosmology: Acceleration from D-cceleration. *Phys. Rev. D*, 70: 103505, 2004. doi: 10.1103/PhysRevD.70.103505.

Sirignano, J. and Spiliopoulos, K. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, Dec 2018. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.08.029. URL http://dx.doi.org/10.1016/j.jcp.2018.08.029.

Susskind, L. and Hrabovsky, G. *The Theoretical Minimum: What you need to know to start doing physics*. Basic Books, USA, 2013.

Szász, D. Multidimensional hyperbolic billiards. *arXiv e-prints*, art. arXiv:1701.02955, January 2017.

Tolley, A. J. and Wyman, M. Equilateral non-gaussianity from multifield dynamics. *Physical Review D*, 81(4), Feb 2010. ISSN 1550-2368. doi: 10.1103/physrevd.81.043502. URL http://dx.doi.org/10.1103/PhysRevD.81.043502.

Weinan, E. and Yu, T. The deep ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6:1–12, 2017.

Yaida, S. Fluctuation-dissipation relations for stochastic gradient descent. In *International Conference on Learning Representations*, 2018.

Yang, G., Hu, E. J., Babuschkin, I., Sidor, S., Liu, X., Farhi, D., Ryder, N., Pachocki, J., Chen, W., and Gao, J. Tensor Programs V: Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer. *arXiv e-prints*, art. arXiv:2203.03466, March 2022.

Zang, Y., Bao, G., Ye, X., and Zhou, H. Weak adversarial networks for high-dimensional partial differential equations. *Journal of Computational Physics*, 411:109409, 2020.

A. Appendix: Further Details of ECD.

Here we collect further details and comments on the theory of ECD and BI. Section A.1 spells out the derivation of the BI dynamics from the framework of Lagrangian and Hamiltonian classical mechanics in physics. In §A.2 we give additional examples of ECD dynamics. Section A.3 supplies details of the calculation of the phase space volume (10). Section A.4 explores the sharp distinctions between the stochastic version of BI and SGDM, finding reduced diffusion in BI due to the speed limit. This enables BBI to explore the landscape with less hard to characterize noise (whose role is replaced by the bounces of BBI).

A.1. BI Dynamics From the Action and Hamiltonian

Optimization is analogous to a physical particle propagating on the loss landscape. In physics, the equations of classical mechanics can be efficiently derived by extremizing an *action functional* $S = \int dtL$ in terms of a *Lagrangian* L. Their first-order form derives from a *Hamiltonian* H. The absence of explicit time-dependence in the S or equivalently in H yields conservation of energy. This formalism may be found in many classical mechanics textbooks, and is succinctly explained without prerequisites in e.g. Lectures 6-8 of (Susskind & Hrabovsky, 2013) or (Landau et al., 1976) Chapters 1,2,7.

We quickly review this formalism in App. A.1.1, with an emphasis on obtaining GDM continuum equations from a *time-dependent* action. There is a relation, known as *Noether's theorem*, between conservation of energy and the absence of explicit time dependence in the specification of the physical theory. Conversely, time-dependence in the specification of the theory can capture the effect of the the dissipation of energy into an additional sector of a larger energy-conserving system. For a discussion of Noether's theorem, see e.g. ((Landau et al., 1976), Chapter 2). In this appendix, we will spell out these results in the case relevant for its relation to optimization dynamics.

After explaining this in the non-relativistic case analogous to GDM, we specialize the formalism to BBI's energy-conserving dynamics in App. A.1.2 with a *time-independent* action and briefly introduce other ECD for optimization in App. A.2.

A.1.1. Actions, Hamiltonians and Gradient Descent with Momentum

If a mechanical system is identified by a vector of position $\boldsymbol{\Theta}$ and its velocity $\dot{\boldsymbol{\Theta}}$, where the dot denotes a time-derivative, its dynamics (meaning the equations governing its time evolution starting from an initial point $\boldsymbol{\Theta}_0$ with initial velocity $\dot{\boldsymbol{\Theta}}_0$) can be encoded in an *action functional*

$$S \equiv \int dtL(\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}, t)\,, \tag{20}$$

constructed in terms of a function $L(\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}, t)$ called *Lagrangian*. The time evolution of the system is then determined by the Euler-Lagrange equations

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\boldsymbol{\Theta}}}\right) = \frac{\partial L}{\partial \boldsymbol{\Theta}}. \tag{21}$$

which result from demanding that the variation of the action vanishes (Susskind & Hrabovsky, 2013).

To gain some intuition, let us apply this formalism to the action

$$S_{GDM} = \int dte^{\frac{f}{m}t}\left[\frac{m\dot{\boldsymbol{\Theta}}^2}{2} - V(\boldsymbol{\Theta})\right]\,. \tag{22}$$

The Euler-Lagrange equations (21) give

$$m\ddot{\boldsymbol{\Theta}} = -\boldsymbol{\nabla}V - f\dot{\boldsymbol{\Theta}} \tag{23}$$

This second-order system describes the motion of a particle of mass m under the force $-\boldsymbol{\nabla}V$, slowed down by friction controlled by the coefficient f.

A convenient and conceptually important rewriting of the same physics in a first-order fashion can be obtained by switching to *Hamiltionian* formalism. To do so, one defines the vector of *momenta*

$$\boldsymbol{\Pi} \equiv \frac{\partial L}{\partial \dot{\boldsymbol{\Theta}}} \tag{24}$$

and an *Hamiltonian* function

$$H(\boldsymbol{\Theta}, \boldsymbol{\Pi}, t) \equiv \boldsymbol{\Pi}\dot{\boldsymbol{\Theta}} - L(\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}, t)\,, \tag{25}$$

where all the instances of $\dot{\mathbf{\Theta}}$ in (25) are substituted for $\mathbf{\Pi}$, $\mathbf{\Theta}$ and t using (24). In this language, the Euler-Lagrange equations (21) are translated to Hamilton-Jacobi equations

$$\dot{\mathbf{\Theta}} = \frac{\partial H}{\partial \mathbf{\Pi}} \,, \qquad \dot{\mathbf{\Pi}} = -\frac{\partial H}{\partial \mathbf{\Theta}}. \tag{26}$$

Applying the Hamiltonian formalism to the GDM action (22) transforms the system (23) into

$$\dot{\mathbf{\Theta}} = \frac{\mathbf{\Pi}}{m} \,, \qquad \dot{\mathbf{\Pi}} = -\frac{f}{m}\mathbf{\Pi} - \boldsymbol{\nabla} V. \tag{27}$$

Substituting the first equation here into the second recovers the second order form (23). The system (27) is a continuum version of the GDM algorithm, as discussed recently e.g. in (França et al., 2020; Muehlebach & Jordan, 2021) and references therein. Extra care with the map continuous $\leftrightarrow$ discrete is required when considering the effects of stochasticity added to GDM by noisy gradients (e.g. (Yaida, 2018; Kunin et al., 2021)). We will analyze the effect of noise for BBI in Sec. A.4.

The Hamiltonian is interpreted as the total *energy* E of the system. However, in the GDM example described so far, one can easily check that it not conserved, i.e. its value changes during the evolution:

$$\frac{d}{dt} H_{GDM} \neq 0 \qquad\qquad E \text{ is not conserved.} \tag{28}$$

Physically, energy has been lost due the heat generated by friction. Mathematically, this non-conservation is a consequence of the Lagrangian (22) depending *explicitly* on t.

A.1.2. Energy Conserving Dynamics and BBI

The Born-Infeld (BI) dynamics (3)-(5) can be derived at the continuum level from the action

$$S_{BI} = \int dt L = -\int dt V(\mathbf{\Theta}) \sqrt{1 - \frac{\dot{\mathbf{\Theta}}^2}{V(\mathbf{\Theta})}}. \tag{29}$$

Using (24) we derive the momentum

$$\mathbf{\Pi} = \frac{\partial L}{\partial \dot{\mathbf{\Theta}}} = \frac{\dot{\mathbf{\Theta}}}{\sqrt{1 - \frac{\dot{\mathbf{\Theta}}^2}{V(\mathbf{\Theta})}}} \tag{30}$$

and from (25) the Hamiltonian

$$H_{BI} = \mathbf{\Pi}\dot{\mathbf{\Theta}} - L = \frac{V(\mathbf{\Theta})}{\sqrt{1 - \frac{\dot{\mathbf{\Theta}}^2}{V(\mathbf{\Theta})}}} = \sqrt{V(\mathbf{\Theta})(V(\mathbf{\Theta}) + \mathbf{\Pi}^2)}. \tag{31}$$

From this we extract the continuum equations of motion (3), which we repeat here for convenience of the reader:

$$\dot{\mathbf{\Theta}} = \mathbf{\Pi}\frac{V(\mathbf{\Theta})}{E}, \qquad \dot{\mathbf{\Pi}} = -\frac{\boldsymbol{\nabla} V}{2}\left(\frac{E}{V(\mathbf{\Theta})} + \frac{V(\mathbf{\Theta})}{E}\right), \tag{32}$$

where we defined $E \equiv H_{BI}$. As explained in the main text, a first order symplectic integration of these equations produces the discretized BI evolution.

For the BI dynamics, the Hamiltonian (31) is conserved, meaning that its value is fixed during the evolution generated by (32):

$$\frac{d}{dt} H_{BI} = 0 \qquad\qquad E \text{ is a constant.} \tag{33}$$

From eq. (31) we see that the speed of motion in parameter space $|\dot{\mathbf{\Theta}}|$ cannot exceed V, and conversely that $\dot{\mathbf{\Theta}}^2$ will not vanish for $0 < V < E$. This is very different from gradient descent with or without momentum.

A.2. Other Examples of ECD

There is a large space of physical models with energy conserving, frictionless dynamics that slows to a stop as $V \to 0$. Another mentioned in §1 has action (in the language of §A.1)

$$S = \int dt \left(\frac{1}{2} m(\boldsymbol{\Theta}) \dot{\boldsymbol{\Theta}}^2 \right) = \int dt \left(\frac{1}{2V(\boldsymbol{\Theta})} \dot{\boldsymbol{\Theta}}^2 \right) \tag{34}$$

Here the objective $V \to 0$ occurs when the mass ($= 1/V$) blows up, causing the particle to slow to a stop.

For such a model the first order equations of motion read

$$\dot{\boldsymbol{\Theta}} = \boldsymbol{\Pi} V(\boldsymbol{\Theta}) \qquad \dot{\boldsymbol{\Pi}} = -\frac{E}{V(\boldsymbol{\Theta})} \boldsymbol{\nabla} V \,, \tag{35}$$

descending from the Hamiltonian

$$H = \frac{1}{2} V(\boldsymbol{\Theta}) \boldsymbol{\Pi}^2 . \tag{36}$$

A symplectic integration of the equations (35) produces another example of an optimization algorithm that conserves energy and shares the properties of BBI described in the main text.

Another example analogous to BI and also inspired by a similar theoretical cosmology model (Mathis et al., 2021) has a logarithmic rather than a square root branch cut enforcing the speed limit:

$$S = \int dt \left[-\frac{V(\boldsymbol{\Theta})}{2} \log \left(1 - \frac{\dot{\boldsymbol{\Theta}}^2}{V(\boldsymbol{\Theta})} \right) \right] \tag{37}$$

In all these cases, including BI, the shifted loss function $V(\boldsymbol{\Theta})$ (2) may be replaced by any function of it that approaches 0 as $V \to 0$. In general, there is a high-dimensional space of possible ECD models, all with the same general properties listed in Table 1.

In the context of theoretical cosmology, the classes of models of early universe inflation (accelerated expansion of the universe) which motivated ECD are distinctive in terms of their dynamics and their observational signatures. These enable empirical discrimination between them and the GD-like dynamics of so-called slow-roll inflation. (Armendariz-Picon et al., 1999; Alishahiha et al., 2004; Chen et al., 2007; Cheung et al., 2008). In the present work, we similarly see sharp distinctions between frictional and ECD optimization which will be interesting to further explore and exploit.

A.3. Details in the Calculation of the Phase Space Volume Formula Predicting the Distribution of Results

Here we supply the steps deriving (10) from (38).

The total volume is given by

$$\text{Vol}(\mathcal{M}) = \int d^n \pi d^n \theta \delta(\sqrt{V(V + \boldsymbol{\Pi}^2)} - E) \tag{38}$$

Writing $|\boldsymbol{\Pi}| = \tilde{\pi}$ and doing the angular integral in the momenta gives

$$\text{Vol}(\mathcal{M}) = \frac{2\pi^{n/2}}{\Gamma(n/2)} \int d^n \theta \int_0^\infty d\tilde{\pi} \tilde{\pi}^{n-1} \delta(\sqrt{V(V + \tilde{\pi}^2)} - E) \,. \tag{39}$$

Using the δ function to do the $\tilde{\pi}$ integral then yields

$$\text{Vol}(\mathcal{M}) = \frac{2\pi^{n/2}}{\Gamma(n/2)} \int d^n \theta \frac{E}{V} \left(\frac{E^2}{V} - V \right)^{\frac{n-2}{2}} \,. \tag{40}$$

The analogous formula for pure momentum (without speed limit or friction) would be

$$\text{Vol}(\mathcal{M}) \propto \int d^n \theta (E - V)^{\frac{n-2}{2}} \qquad \text{frictionless non-relativistic momentum} \tag{41}$$

and does not exhibit the dominance at low V of BBI (40). Different forms of ECD will produces analogous formulas to (40), sometimes enhancing the dominance at small V. It will be interesting to explore the performance of such generalizations, including the replacement of V with a more general function $g(V)$ of the loss.

Let us consider the contributions to the integral near the global minimum (or degenerate minima) at some $\mathbf{\Theta}_0$ where $V \to 0$. We can expand the measure,

$$d^n\theta = |\mathbf{\Theta} - \mathbf{\Theta}_0|^{n-1} d|\mathbf{\Theta} - \mathbf{\Theta}_0| d\Omega_{\mathbf{\Theta}-\mathbf{\Theta}_0} \tag{42}$$

and also expand $V \propto |\mathbf{\Theta} - \mathbf{\Theta}_0|^\kappa$ for some κ. From (40) we see that the singularity there is integrable only for $\kappa < 2$. One would expect $\kappa = 2$ for a smooth potential V with global minimum $V_{\text{global}} = 0$. This leads to a logarithmically divergent volume as we approach V_{global}.

More generally, we can compare the phase space volume in the basins of attraction of different local minima. Starting from (40) and working in the regime of $V \ll E$, we can write the measure near the Ith local (or global) minimum as

$$\text{Vol}(\mathcal{M}_I) = \frac{2\pi^{n/2}}{\Gamma(n/2)} E^{n-1} \int d^n(\theta - \theta_I) V^{-n/2} \tag{43}$$

Near a minimum, V is quadratic, giving (after an orthogonal diagonalization of its Hessian)

$$V \simeq V_I + \frac{1}{2}\sum_{i=1}^{n} m_{Ii}^2 (\theta_i - \theta_{Ii})^2 \ . \tag{44}$$

Defining

$$\eta_{iI} = m_{iI}(\theta_i - \theta_{Ii}) \Rightarrow V \simeq V_I + \frac{1}{2}\sum_{i=1}^{n} \eta_{Ii}^2 \ , \tag{45}$$

we get

$$\text{Vol}(\mathcal{M}_I) = \frac{2\pi^{n/2}}{\Gamma(n/2)} \frac{E^{n-1}}{\prod_i m_{Ii}} \int d^n\eta V^{-n/2} = \left(\frac{2\pi^{n/2}}{\Gamma(n/2)}\right)^2 \frac{E^{n-1}}{\prod_i m_{Ii}} \int d\eta \frac{\eta^{n-1}}{(V_I + \frac{1}{2}\eta^2)^{n/2}} \ , \tag{46}$$

where $\eta \equiv |\eta_I|$ exhibiting a logarithmic divergence as $V_I \to 0$ as in (10).

Integrating η from 0 to 1 here, as an estimate of the contribution to the measure of this Ith minimum, gives a formula for the volume in terms of a hypergeometric function:

$$\text{Vol}(\mathcal{M}_I) = \left(\frac{2\pi^{n/2}}{\Gamma(n/2)}\right)^2 \frac{E^{n-1}}{\prod_i m_{Ii}} \frac{V_I^{-n/2}}{n} \ {}_2F_1(\frac{n}{2}, \frac{n}{2}, \frac{n}{2}+1, -\frac{1}{2V_I}) \ . \tag{47}$$

A.4. Further Study of BI in the Stochastic Case

As noted in the main text, it is often necessary to separate the data (or the input points sampled in solving differential equations) into minibatches x_B, switching from batch B to batch $B+1$ at time t_B, i.e. switching to a new minibatch every $\Delta(t_{B+1} - t_B)/\Delta t$ steps. As explained in §3.2, the essential features of ECD persist in this case, as is borne out by the experiments.

In this section we explore in a little more detail the stochastic behavior of BI. Similarly to the noise-free case, we find clear distinctions between this class of optimizers and friction-based ones such as SGDM. It would be interesting to extend these preliminary studies in the future. We note interesting prior work (Tolley & Wyman, 2010) on stochastic effects in the DBI theory of early universe inflation (Alishahiha et al., 2004).

The transitions to new batches means that at a given time, the algorithm evolves in the loss landscape given by $V^B = V(\mathbf{\Theta}; x_B)$ rather than the full loss $V(\mathbf{\Theta}; x)$. The loss therefore behaves like a time dependent potential in the corresponding Hamiltonian system. In this physical system, that would generally lead to energy non-conservation by an amount determined by the strength of the time dependence, $(V^{B+1} - V^B)/V^B \equiv \Delta V_B/V^B$. In our algorithm, however, the situation is somewhat different since we keep the energy fixed via the rescalings described in the main text.

To begin let us describe the rescaling required to preserve energy E in the presence of minibatches (even in the absence of discretization error). From (31) we have $\mathbf{\Pi}^2 = \frac{E^2}{V} - V$. Thus as we transition from batch B to batch $B+1$, we have to rescale by a factor

$$\lambda = \frac{\pi_{i,B+1}}{\pi_{i,B}} = \sqrt{\frac{E^2/V^{B+1} - V^{B+1}}{E^2/V^B - V^B}} \tag{48}$$

in order to conserve energy.

Before commenting on the small learning rate regime, we first note that the prescription of (Yaida, 2018), deriving averaged correlations of observables under the assumption of a late time steady state distribution, extends to the BI case. Following (Yaida, 2018), if we assume an equilibrium distribution at late times, we can compute appropriate correlation functions in the putative distribution. Those arising at the quadratic order in $\mathbf{\Theta}, \mathbf{\Pi}$ are:

$$\langle [|\frac{\partial_i V^B}{V^B}\theta_j|]\rangle = \langle [|\frac{\partial_j V^B}{V^B}\theta_i|], \;\; \langle\, V(\theta_i\pi_j + \theta_j\pi_i)\,\rangle = \Delta t E\,\langle\, \frac{1}{4}(\partial_i V\theta_j + \partial_j V\theta_i) - [|\frac{(V^B)^2}{2E^2}\pi_i\pi_j|]\,\rangle \tag{49}$$

where as in that work, V^B denotes the loss function for batch B, $[|\dots|]$ represents a minibatch average, whereas $\langle\dots\rangle$ represents the expectation value in the steady state distribution. Here we assumed $V^B \ll E$ to simplify the formulas. The last relation here is comparable to equation (28) in (Yaida, 2018), with the similarity clearer upon noting from (30)-(31) that the velocity is given by $v_i = \dot\theta_i = \pi_i V/E$. Although the relations look similar, they encode very different behavior: Whereas in SGDM $\langle v^2\rangle$ is generally nonzero (and in ordinary Brownian motion it is approximately constant), in BI this quantity is bounded by V because of the loss-dependent speed limit.

Next, we briefly study the small Δt regime. By plugging the second equation in (3) into the first and taking into account the additional time-dependence in V and in π from the rescaling (48), we obtain the second-order equations

$$\ddot\theta_i = -\frac{1}{2}\partial_i V + \dot\theta_i(\frac{\dot{\mathbf{\Theta}}\cdot\nabla V}{V}) + \dot\theta_i \sum_B \delta(t-t_B)\left(1 - \lambda(\Delta V^B) + \frac{\Delta V^B}{V^B}\right). \tag{50}$$

For the present discussion we do not include the billiards-inspired bounces prescribed in the algorithm. All of the terms multiplying $\delta(t-t_B)$ are of order $\Delta V^B/V^B$ and may be expected to ensemble-average to zero (including when multiplied by other quantities such as $\dot\theta_i$ that are not tied to the random choice of new batch). We could have obtained the resulting equation equally well from our discretization of BI by plugging (5) into (6).

We may contrast this to a similar small learning-rate limit (Kunin et al., 2021) for SGDM:

$$\frac{\Delta t}{2}(1+\beta)\ddot{\mathbf{\Theta}} + (1-\beta)\dot{\mathbf{\Theta}} = -\nabla V^B \tag{51}$$

derived from the update rule

$$\mathbf{v}_{k+1} - \beta\mathbf{v}_k = -\nabla V_k, \quad \mathbf{\Theta}_{k+1} - \mathbf{\Theta}_k = \Delta t\, \mathbf{v}_{k+1}\,. \tag{52}$$

We note the absence of a Δt in the first update rule, which is standard in machine learning but different from the standard discretization of the analogous non-relativistic classical particle model and hence different from the non-relativistic regime of our discretized BI model.

Aside from the distinction in the placement of Δt factors, we note the presence of a friction term in (51) and its absence in (50). In both cases, the batch dependence in $V(\mathbf{\Theta}, t)$ constitutes a source of noise, in general with nontrivial statistics.

For now, we contrast the form that Brownian motion takes in the two systems in this small-learning-rate regime. According to (Kunin et al., 2021), SGDM exhibits anomalous Brownian motion, with the variance of the motion behaving as $\langle\mathbf{\Theta}^2\rangle \propto t^\kappa$ for some positive exponent κ. The minimal version of Brownian motion in physics can be derived from the Langevin equation $\ddot\theta + \gamma\dot\theta = \xi$ [10] where the noise term satisfies $\langle\xi(t)\xi(t')\rangle = \delta(t-t')$, which is very loosely similar to (51), leading to the relation $\frac{d}{dt}\langle\theta^2\rangle = 2\langle\dot\theta^2\rangle/\gamma = \text{constant}$.

In comparing to (50), although there is no friction term, we can see that for a potential basin of the form $\frac{1}{2}m^2\mathbf{\Theta}^2$, for speed-limited motion along the gradient direction we have $\langle\dot{\mathbf{\Theta}}\cdot\nabla V/V\rangle \sim -1 + \text{fluctuations}$. Comparing to the standard

[10] See e.g. https://web.stanford.edu/ peastman/statmech/friction.html .

Brownian motion result just summarized, we again expect $\frac{d}{dt}\langle \boldsymbol{\Theta}^2 \rangle \sim 2\langle \dot{\boldsymbol{\Theta}}^2 \rangle$. But in contrast to that case and SGDM, for BI the speed limit strongly constrains the right hand side. Thus we expect much less diffusion for BI. Instead, for BI it is the billiards-inspired bounces lead to mixing over the phase space. Clearly it would be interesting to flesh out these distinctions in more detail, along with their implications for representation learning in ML and for coverage of PDE solutions.

B. Details on the PDE Problems

Here we describe in some details the class of PDE problems on which we tested the BBI algorithm.

We considered the class of nonlinear Poisson Dirichlet problems defined by

$$\text{find } u \text{ such that} \qquad \begin{cases} \Delta u + u^2 = f & x \in \Omega \\ u = f_0 & x \in \partial\Omega \end{cases}, \tag{53}$$

where Ω is a domain in $\mathbb{R}^d$ with boundary $\partial\Omega$, $\Delta = -\sum_{i=1}^d \partial^2_{x_i}$, and f is a known function.

To design a specific problem we then proceeded backwards: we chose a certain u with a non-trivial shape, plugged it on the left hand sides of (53) and computed f (and f_0). This allowed us to know analytically one of the solutions of the problem. Specifically, for the experiment presented in Sec. 4.4 we worked in the unit-ball in $\mathbb{R}^2$ and considered as analytic solution the function

$$u_{\text{an.}} = \sin^2(20(x_1^2 + x_2^2)) \tag{54}$$

of which we plot a section in Fig. 7 (left).

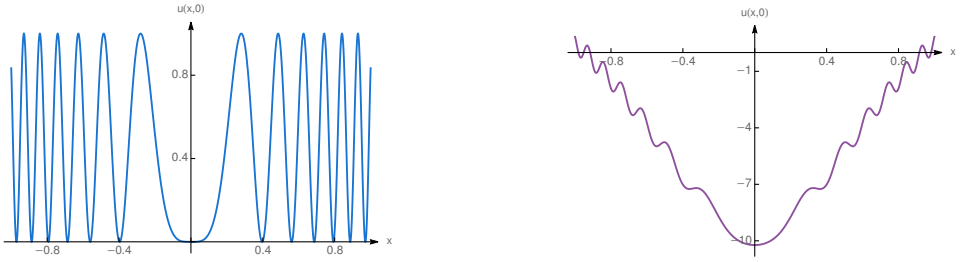

Figure 7. The solutions to the PDE problem studied in Sec. 4.4. On the left a section at $x_2 = 0$ of the analytic solution (54); on the right, a section of the other solution determined numerically with a shooting method.

We have chosen such a wiggly solution in order to obtain a relatively hard numerical problem. Plugging it in Prob. (53) , we extract f and f_0:

$$f = \frac{1}{8}\left(3 - 4(1 + 6400(x_1^2 + x_2^2))\cos(40(x_1^2 + x_2^2)) + \cos(80(x_1^2 + x_2^2)) - 640\sin(40(x_1^2 + x_2^2))\right) \tag{55}$$

and $f_0 = 0.833469$. The problem (54) being nonlinear, we can ask whether other solutions exist. As can be shown by employing radial symmetry together with a shooting method, the answer is affirmative: two solutions exist with second one represented in Fig. 7 (right). We stress that spherical symmetry and the reduction to a one-dimensional radial equation are used only to employ a standard method to assess the existence of other solutions, but has not been employed in any way in the solution of Prob. (53) with NNs, since ultimately the goal is to being able to solve higher-dimensional problems with no symmetry assumptions.

For the experiments in Sec. 4.4, we then used as an ansatz for solving (53) a residual network defined as

$$\begin{aligned} y^i_1 &= \sigma(\textstyle\sum_{\mu=1}^n W_0^{i\mu} x_\mu + b_0^i) \\ y^i_{s+1} &= y^i_s + \sigma(\textstyle\sum_{j=1}^N W_s^{ij} y_s^j + b_s^i) \\ y_{D+2} &= \textstyle\sum_{j=1}^N W_{D+1}^j y_{D+1}^j + b_{D+1} \end{aligned} \tag{56}$$

where $s = 1, \dots D$ and $i = 1, \dots, N$ with D and N being respectively the depth and width. d is the dimension of the PDE problem, and in our examples restrict to $d = 2$. As *activation function* σ we chose the logistic sigmoid. We briefly experimented with other smooth activation functions (tanh, sin) and not observing degradation of performance, we focused

on the sigmoid to show the distribution of results. The sets of weight and biases, W and b, define the set of parameters collectively denoted as $\mathbf{\Theta}$ in the main text. Summing up, the sequence of transformations in (56) is defining a function from $\mathbb{R}^d \to \mathbb{R}$, depending on these parameters, which is used as an ansatz for the PDE solution. Schematically,

$$\mathcal{N}(x; \mathbf{\Theta}) \equiv y_{D+2}(y_{D+1}(\dots)). \tag{57}$$

In our experiments we used 3 middle layers of width 200, i.e. $D = 3$ and $N = 200$. The loss function (19) is then constructed with an L^2 regularization term $\mathrm{R}(\mathbf{\Theta}) = 2 \times 10^{-4}\mathbf{\Theta}^2$, $\gamma = 10^5$ and by sampling respectively 10^4 and 10^3 random points from the unit ball in $\mathbb{R}^2$ and its boundary.