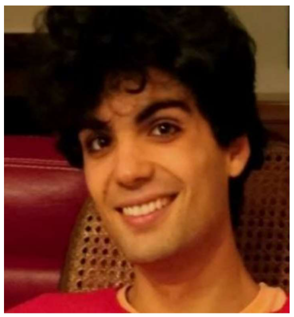
Born-Infeld (BI) for AI:

Energy-Conserving Descent (ECD) for Optimization



Stanford Institute for Theoretical Physics (SITP)

G. Bruno De Luca*

Eva Silverstein*

Acknowledgements ⊃

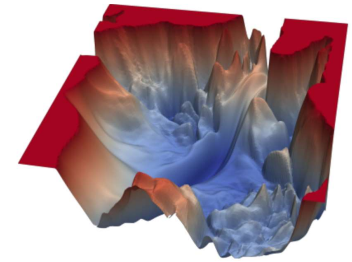
Discussions w/ J. Batson, Y. Kahn, D. Roberts on inflationary cosmology and optimization; early collaboration G. Panagopoulos, T. Bachlechner

New discussions/collaborations: ML (Kunin), Computational Chemistry (Zhang), Sampling (Robnik/Seljak), …

Reviewers & conference organizers

Overview: Optimization of an objective function F

- Data analysis/Machine Learning [F = loss]
- Solving (Partial) Differential Equations
  [F = Σ (PDEs)^2+(boundary conditions)^2]
- Many scientific applications



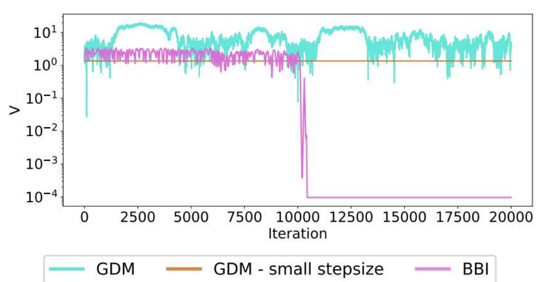
[Image from Li et al. , '18]

Gradient Descent with Momentum (GDM) can work well with modern tweaks.
Physical analogue: particle motion on potential energy V = F, *with friction*, discretized.

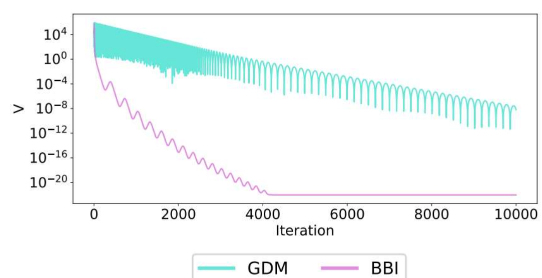
Our proposal: Energy Conserving Descent (**ECD**): discretized physical evolution, *without friction*, nonetheless slowing near minimal F. Examples include:

- BBI: relativistic, (speed limit)^2 = V = F-ΔV [or more general (speed limit)^2= g(V)]
- Ruthless: non-relativistic, mass ∝ 1/g(V)

Ackley 2d (nonconvex)



Zakharov 10d (shallow)



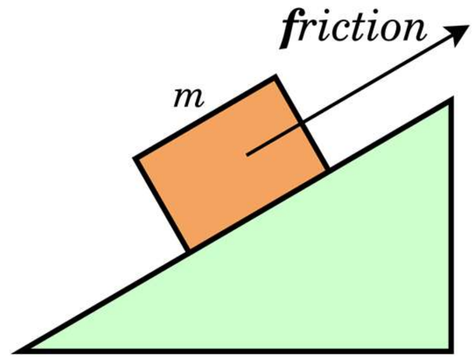
+ other synthetics, PDEs, small ML (Cifar, MNIST, Tiny ImageNet [new]), chemistry, sampling [new]

No friction ⟹ **Energy Conservation** ⟹ favorable properties and improved calculability: concrete formula for distribution of results: in all dims weighted toward small V = F-ΔV

Physics of GDM

Particle descending a potential energy landscape V

$$V(\Theta) = F(\Theta) - \Delta V$$



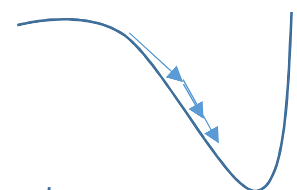
Familiar law of motion: Force = mass × acceleration

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

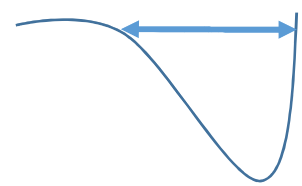
Friction coefficient $\boldsymbol{f}$ $\Longrightarrow$ Energy not conserved

First-order form: $p = m\dot{\Theta} \quad \dot{p} = -p\dfrac{\boldsymbol{f}}{m} - \nabla V$

Discretization → GD with Momentum (GDM) + minibatches → SGDM



- Energy $E = \dfrac{p^2}{2m} + V(\Theta)$ not conserved because of friction



- $\boldsymbol{f} = 0$ would conserve energy, but the particle flies quickly past V≃0, spending very little time there (especially in high dimensions)

ECD: physical dynamics can conserve energy yet slow near V=0

Next: explicit realizations

Explicit realizations of ECD

Change the dynamics to conserve Energy E and favor V≃ 0

General

$$H(\Theta, \Pi) = E$$

Position vector (parameters)

Momentum vector

Dynamical equations (cf. Newton's laws of motion): $\dot{\Theta} = \frac{\partial H}{\partial \Pi}, \qquad \dot{\Pi} = -\frac{\partial H}{\partial \Theta}$

1. **BI:** (speed limit)^2 = V = F-ΔV, [or general function g(V)]

$$H = \sqrt{g(V)\left(\Pi^2 + g(V)\right)} = g(V)/\sqrt{1 - \frac{\dot{\Theta}^2}{g(V)}}$$

- Cannot exceed relativistic speed limit: $\dot{\Theta}^2 \leq g(V)$ [ES, Tong, Alishahiha '04, cf. França et al. '20]

2. **Rootless (Ruthless):** mass ∝ 1/g(V) $H = \left(\frac{\Pi^2}{2m(V)}\right) = \mathrm{g(V)}\ \Pi^2 = \frac{1}{2} m(V)\dot{\Theta}^2$

- Slows as the particle gets heavy: $m(V) \to \infty, \quad g(V) \to 0 \Rightarrow \dot{\Theta}^2 \to 0$

Building ECD optimization algorithms

0. Choose the continuum dynamical system

1. Discretize the continuum equations of motion
   - e.g. BI with g(V) = V:

$$\sqrt{V(V+\vec{\pi}^2)} \equiv E$$

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

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

2. Choose an initialization
   - Common choice: Π(0) => E = V(0)
   - Option: E > 0 => choice of Π(0) compatible with Energy eq.

3. Use discretized equation as update rules

4. Add other features
   - Enforce strict Energy conservation rescaling Π
   - Adaptive tuning of shift ΔV = F-V (next page)
   - Option: random rotation of momenta ("bouncing", explained later)

5. Test it!

Data set	SGD	BBI
MNIST	99.166 , 98.160	99.177 , 99.190
CIFAR-10	92.628 , 92.655	92.434 , 92.435

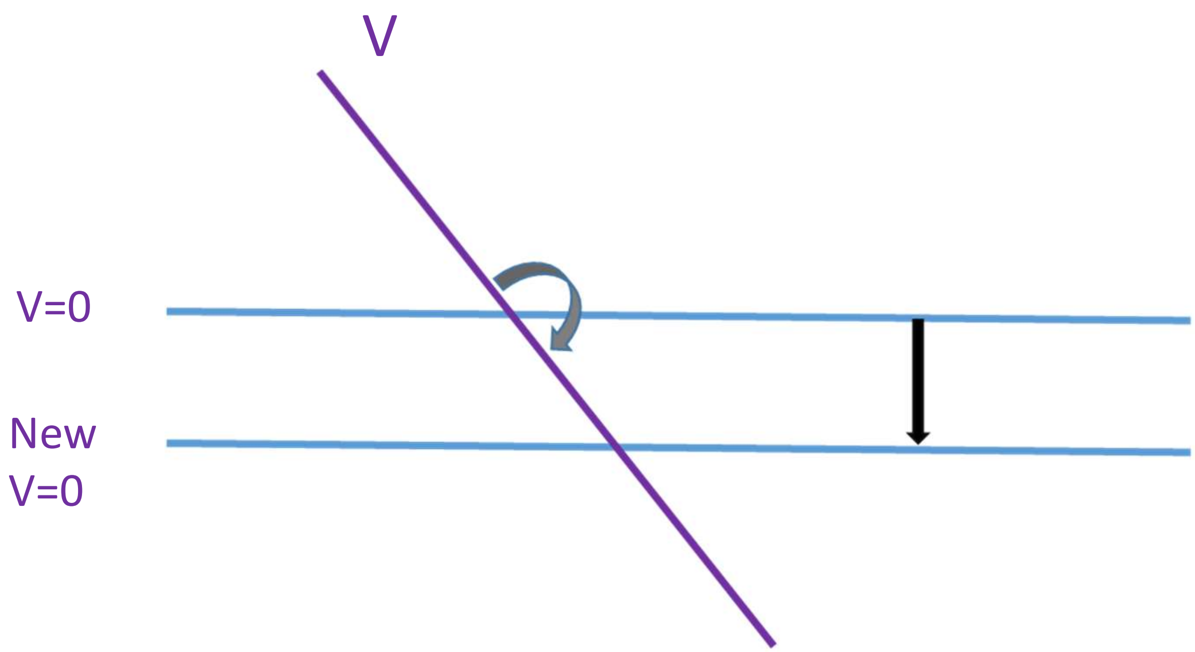
Modest (~50) statistics and limited hyper-parameter tuning (without all the tweaks on either side); just a check of basic competence. "Bouncing" not required here.

Automatic (adaptive) Tuning of ΔV

The value of the loss function F at the objective is not always known:

$$\boxed{V = F - \Delta V}$$

ΔV is a hyperparameter that can automatically adjust (recover from an over-estimate). New upgrade to optimizer code.



Given a too-high initial guess for ΔV, 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, iteratively tuning it.

Recap so far:

- Optimization of an objective function F
- Descent dynamics as (discrete) physical evolution on a potential V = F-ΔV
- Equations of motion (update rules) obtained from a Hamiltonian H
  - Gradient Descent with Momentum: a time-dependent H(Π, Θ, t)
    - Energy not conserved: $\dot{E} = -f\frac{\Pi^2}{m^2} \leqslant 0$
    - Simply removing friction (f =0) does not converge
- Alternative physics: Energy Conserving dynamical systems converging to V→0

$$E = H_{\mathrm{ECD}}(\Theta, \Pi)$$

  - Energy is conserved: $\dot{E} = 0$
  - 2 explicit examples: **BI** [relativistic], **Ruthless** [m = 1/g(V)].
- Discretization gives update rules → new optimization algorithms

Simple benchmarks show that the idea works: friction not needed for optimization.

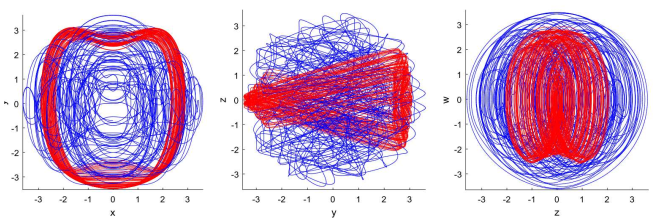
Next: advantages of conserving energy

Energy Conservation

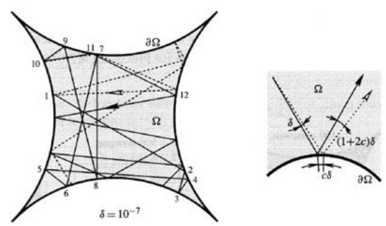
$$H = \frac{V}{\sqrt{1-\frac{\dot{\vec{\theta}}^2}{V}}} = \sqrt{V(V+\vec{\pi}^2)} \equiv E = constant \quad \Longrightarrow$$

- **Cannot** stop unless V=E or V=0, so cannot stop in high local minimum

Can get stuck in orbit at high V. Generically such orbits are unstable: chaos – sensitive dependence on initial conditions – is typical in physical systems. Nearby trajectories disperse roughly on a *mixing* timescale.



[Image from Dong, Yuan, Du et al. '19]



[Image from Encyclopedia of Nonlinear Science, '04]

Chaos and *mixing* has been **proven** in mathematical billiards problems.

This inspires optional Bounces in BI algorithm above to reduce the mixing time $\Longrightarrow$ **BBI**

- *Phase space* (positions & momenta) *volume* is preserved under the evolution. $Vol(phase\ space) = \int d^n\Theta d^n\Pi\,\delta(H(\Pi,\Theta)-E)$

- Past the mixing time, the probability to find a particle from a droplet (bundle of trajectories) in a region M of phase space is ∝ Vol(M)

For ECD, phase space volume is strongly dominated near V=0:

$$Vol(\mathcal{M}) = \frac{2\pi^{n/2}}{\Gamma(n/2)} \int d^n\theta \int d\tilde{\pi}\tilde{\pi}^{n-1}\delta(\sqrt{V(V+\tilde{\pi}^2)} - E) = \frac{2\pi^{n/2}}{\Gamma(n/2)} \int d^n\theta \frac{E}{V}\left(\frac{E^2}{V} - V\right)^{\frac{n-2}{2}}$$

For $V \to 0$, $\quad Vol \propto \int \frac{d^n\Theta}{V^{n/2}} = \int d\Omega \int d|\Theta||\Theta|^{n-1}\frac{1}{V^{n/2}}$

For a basin V $\sim |\Theta|^2$, this becomes $\sim \int d\Omega \int d|\Theta|/|\Theta|$

$V \to g(V) \sim V^\eta \quad \eta > 1$ enhances the preference for V=0 (beats the effect of high dimension n!)

(g(V) also useful for sampling, in addition to optimization)

[GBDL, Roblik, Seljak, ES in progress]

- In contrast, pure momentum would not favor small V:

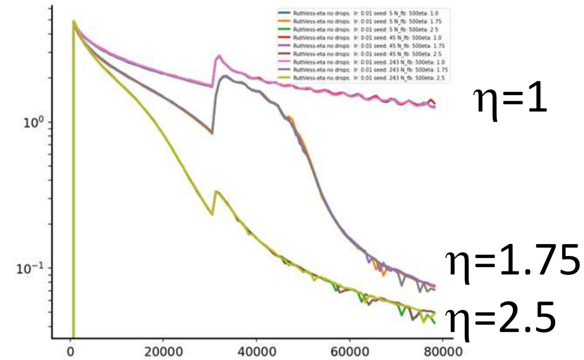
$$\mathrm{Vol}(\mathcal{M}) \propto \int d^n\theta (E - V)^{\frac{n-2}{2}} \qquad \text{frictionless non-relativistic momentum}$$

- The volume formula would not apply at all with friction (less predictive in that sense).

Exploiting the volume formula for image classification (preliminary)

- Enhancement of volume density for η>1 near a quadratic minimum $V \sim \theta^2$:

$$\text{vol} \propto |\Theta|^{n(1-\eta)-1} d|\Theta|$$



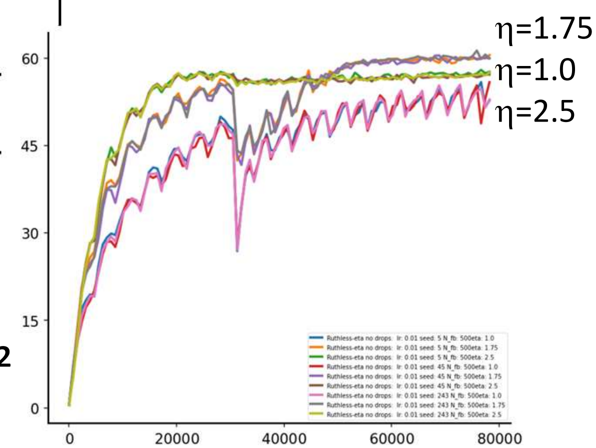
- Small Tests on Tiny-ImageNet* with D. Kunin
  (+ImageNet 1K in progress)

Protocol: lr = 0.01, **no** lr drop needed, 500 bounces,

Averaging of late-epoch weights (SWA) [Izmailov et al. '19]

Training loss decreases monotonically with η, improving test accuracy for intermediate η>1

$m=1/V^\eta$	Accuracy	Accuracy (weights averaged)
η=1	55.44	62.12
η=1.75	61.3	64.1



Compared with SGD: with lr drops

(start 0.1, drop factor 0.1@ep. [30,60,80]) :

Accuracy: 62.52, Accuracy (weights averaged): 62.93

SGD: without lr drops is worse, as well as with loss → loss²

[ECD also > best comparable SGDM in cf. Li et al. '21, Tanaka, Kunin et al. '20...]

*ResNet-18, epochs: 100, batch size: 128, weight decay: 10^{-4}, loss: Cross Entropy

Testing the volume formula

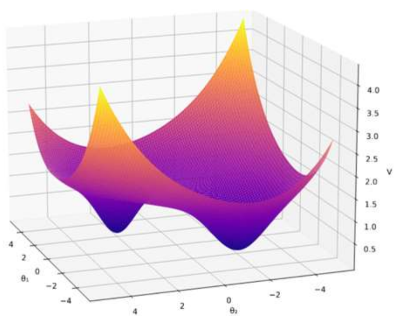
Evaluated in different regions predicts distribution of results (given mixing)

For g(V) = V: $$Vol(\mathcal{M}_\mathcal{I}) = \frac{2\pi^{n/2}}{\Gamma(n/2)} E^{n-1} \int d^n(\theta - \theta_I) V^{-n/2}$$

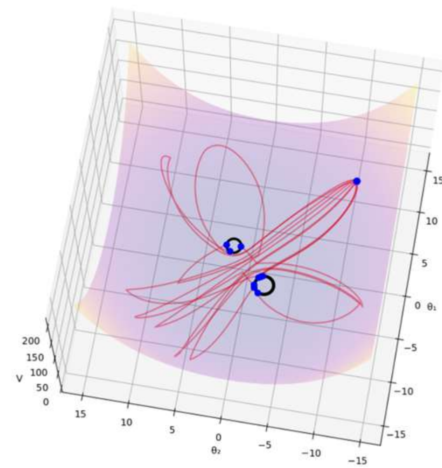
Near a minimum:

$$V \simeq V_I + \frac{1}{2}\sum_{i=1}^{n} m_{Ii}^2 (\theta_i - \theta_{Ii})^2 \qquad Vol(\mathcal{M}_\mathcal{I}) \to b_n \left(\frac{2\pi^{n/2}}{\Gamma(n/2)}\right)^2 \frac{E^{n-1}}{\prod_i m_{Ii}} \log(V_I) \qquad V_I \to 0$$

Empirical check:

V = 

Bouncing trajectories find the 2 basins:



Prediction of ratio of convergence:

$$\frac{\mathrm{Vol}(\mathcal{M}_1)}{\mathrm{Vol}(\mathcal{M}_2)} \sim 1.93$$

Results:

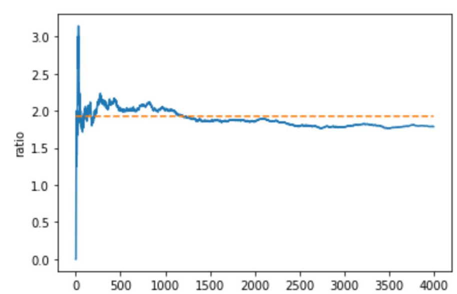


Figure 4: Partial ratios.

Agreement within 10%

Behavior in shallow regions

Volume formula prefers flatter minima

ML lore: flatter minima generalize better

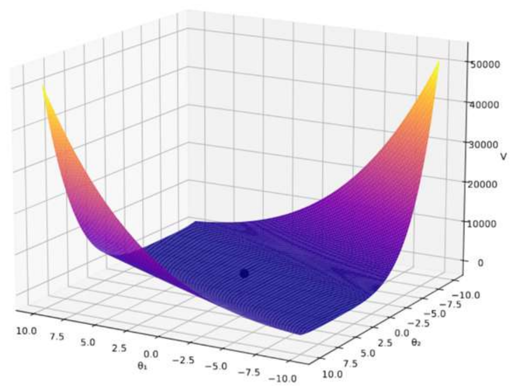
$$V \simeq V_I + \frac{1}{2}\sum_{i=1}^{n} m_{Ii}^2(\theta_i - \theta_{Ii})^2$$

$$Vol(\mathcal{M}_{\mathcal{I}}) \to b_n \left(\frac{2\pi^{n/2}}{\Gamma(n/2)}\right)^2 \frac{E^{n-1}}{\prod_i m_{Ii}} \log(V_I) \quad V_I \to 0, \qquad m_{iI}^2 \to 0$$

Prediction: BI is faster on shallow directions than GD

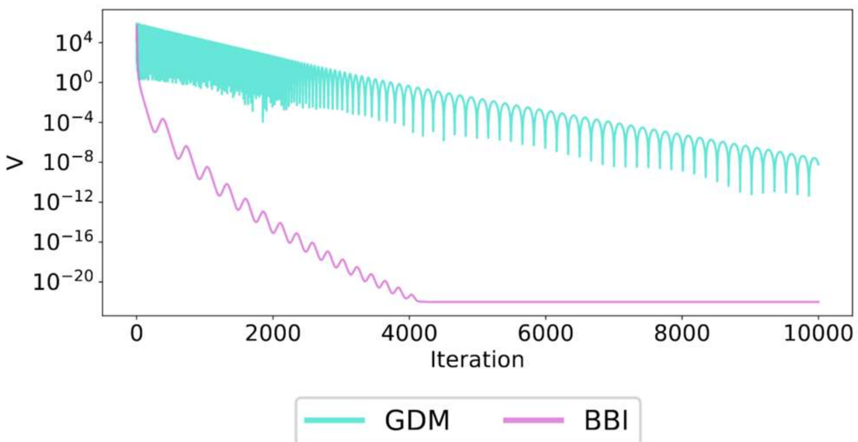
$$\Theta \sim e^{-mt/\sqrt{2}} \quad \text{vs} \quad \Theta \sim e^{-m^2 t/f}$$

Empirical check:



V = 10-dimensional Zakharov function

Results:



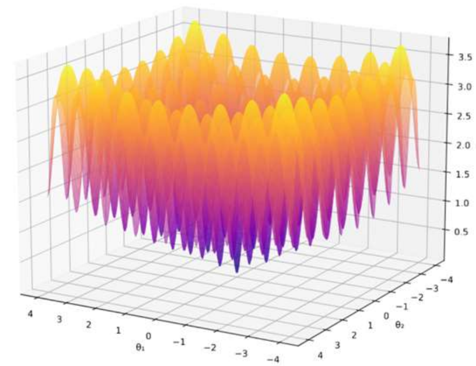
Hyperparameters tuned with hyperopt

Avoiding high local minima

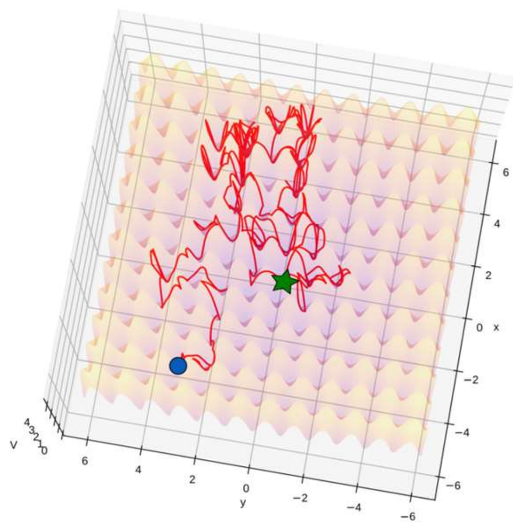
Energy conservation: **ECD** cannot stop in high local minima

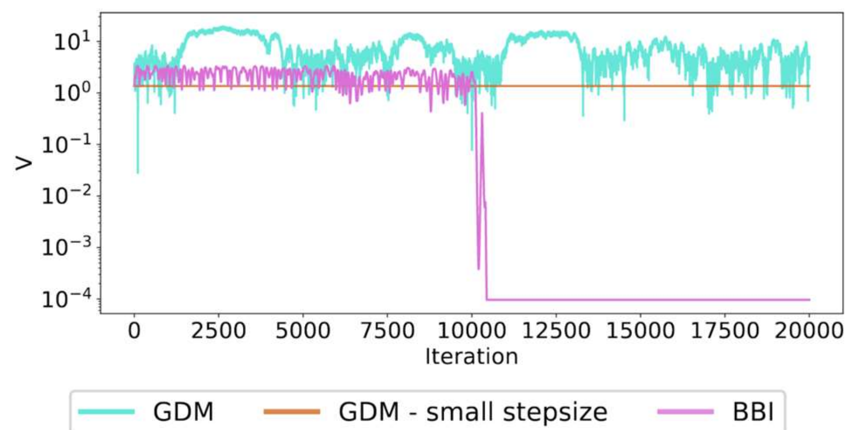
Empirical check: Highly non-convex function

V = 2-dim Ackley function :



Results:





BBI explores and finds the global minimum

Hyperoptimized fixed lr, and for GDM also momentum. GDM either stuck in initial basin or helped out by `catapult' mechanism [Lewkowycz et al. '20], , then more erratic (not settling in global minimum).

Summary comparison

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: $\theta\sim e^{-mt/\sqrt{2}}$	On shallow region: $\theta\sim e^{-m^2t/f}$
Analytic prediction for distribution	Stochastic intuition for distribution
Generalizes	Generalizes

Generalization ok:
speed limit kicks in for
V ≪ E, Vol(phase space)
favors flat basins.

Statements persist with noise (mini-batches) in our prescription:
BBI speed limit tamps down noise, while the bounces (when needed) provide controlled stochasticity for short mixing time.

Application: Solving Partial Differential Equations

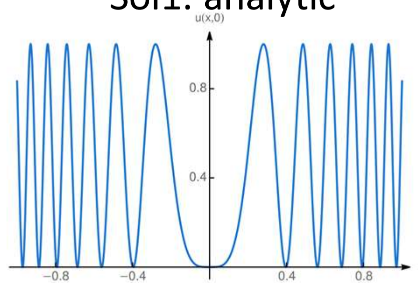
- Most common strategy with ML tools: a NN as ansatz for the PDE: [Lagaris et al. '98, ..., Raissi et al. '19,..]

$$F = V = \sum_{x \in \text{domain}} \text{PDE}[\mathcal{N}(x;\Theta)]^2 + \gamma \sum_{x \in \text{boundary}} \text{BC}[\mathcal{N}(x;\Theta)]^2 + R(\Theta)$$

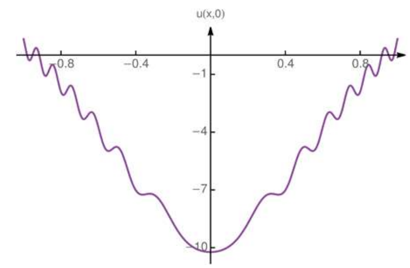
- We reverse-engineered hard (highly nonlinear) 2d PDEs with known multiple solution and checked if ECD optimization finds them

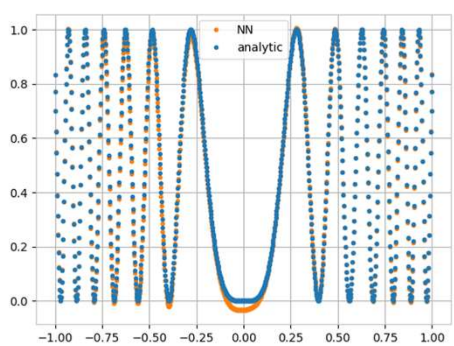
1d slices of *known* solutions:

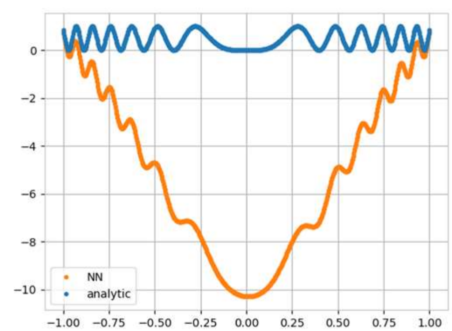
Sol1: analytic



Sol2: numerical



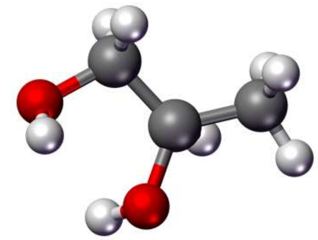




1d slices of *learned* solutions

Found **both** from same initialization: bounces distribute results (mixing)

Ongoing work:



- Quantum Chemistry (with Zhang)
  - Find the minimum energy configuration of a molecule
    F = binding energy < 0 $\implies$ requires ΔV
    - Automatic tuning tested successfully

- Larger scale Machine Learning experiments (with Kunin)
  - Exploit the volume formula from frictionless dynamics for better generalization

- Efficient sampling from a function exp(-F) (with Robnik, Seljak)
  - Reverse engineer g(V) such that

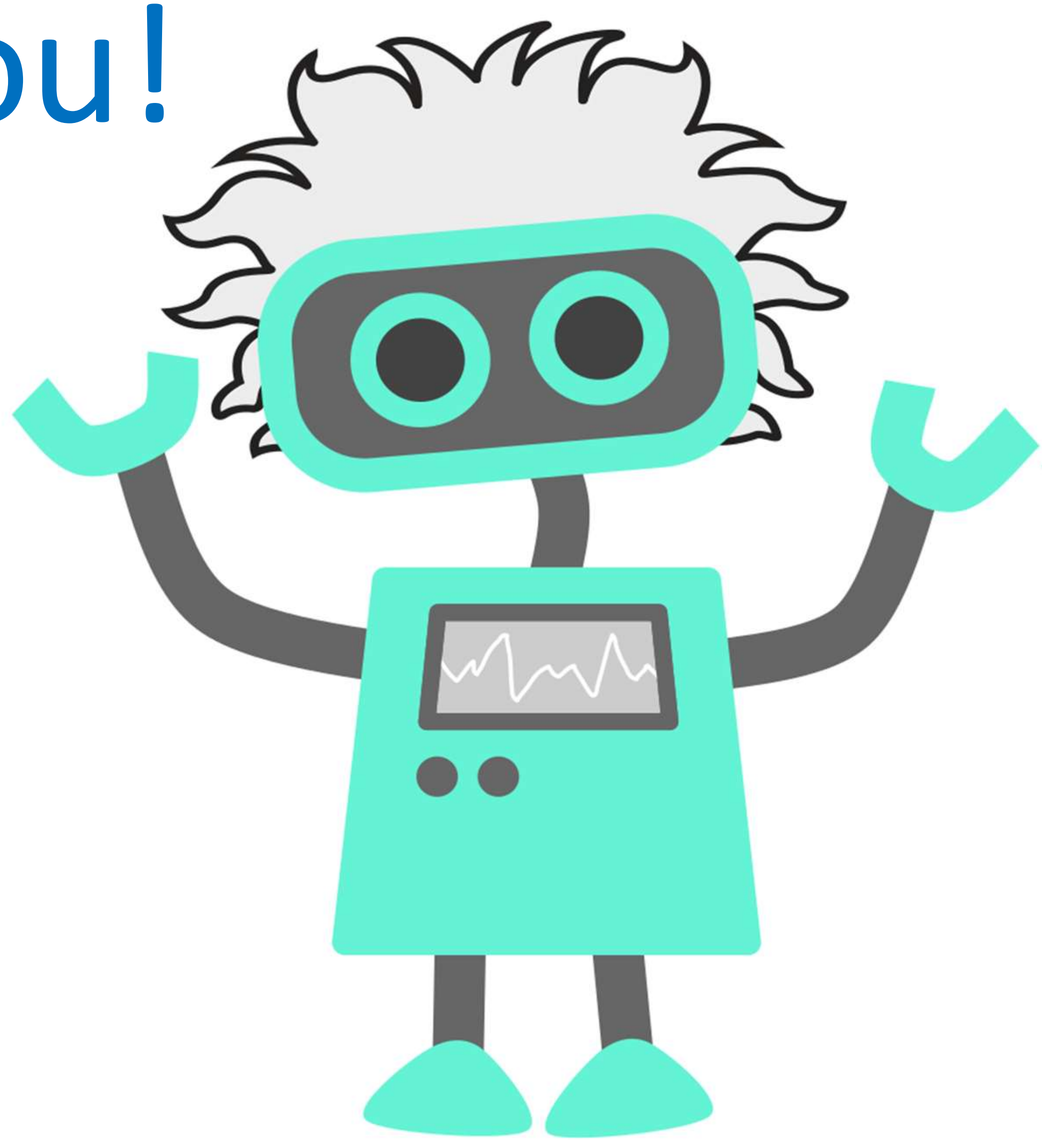
    Vol(phase space)= $\int d^n\Pi \int d^n\Theta \, \exp(-\mathrm{F})\delta\big(E - H(\Theta,\Pi)\big) \propto \int d^n\Theta \exp(-F)$

  - In contrast to Hamiltonian Monte Carlo, no momentum sampling needed

Future directions:

- Feature learning theory and experiment
  - Bounces along the directions of hidden layer parameters

Thank you!



[Robot: publicdomainvectors.org,
Hair: He, Shuhan. (2020). Albert Einstein. Zenodo.
https://doi.org/10.5281/zenodo.3926055]