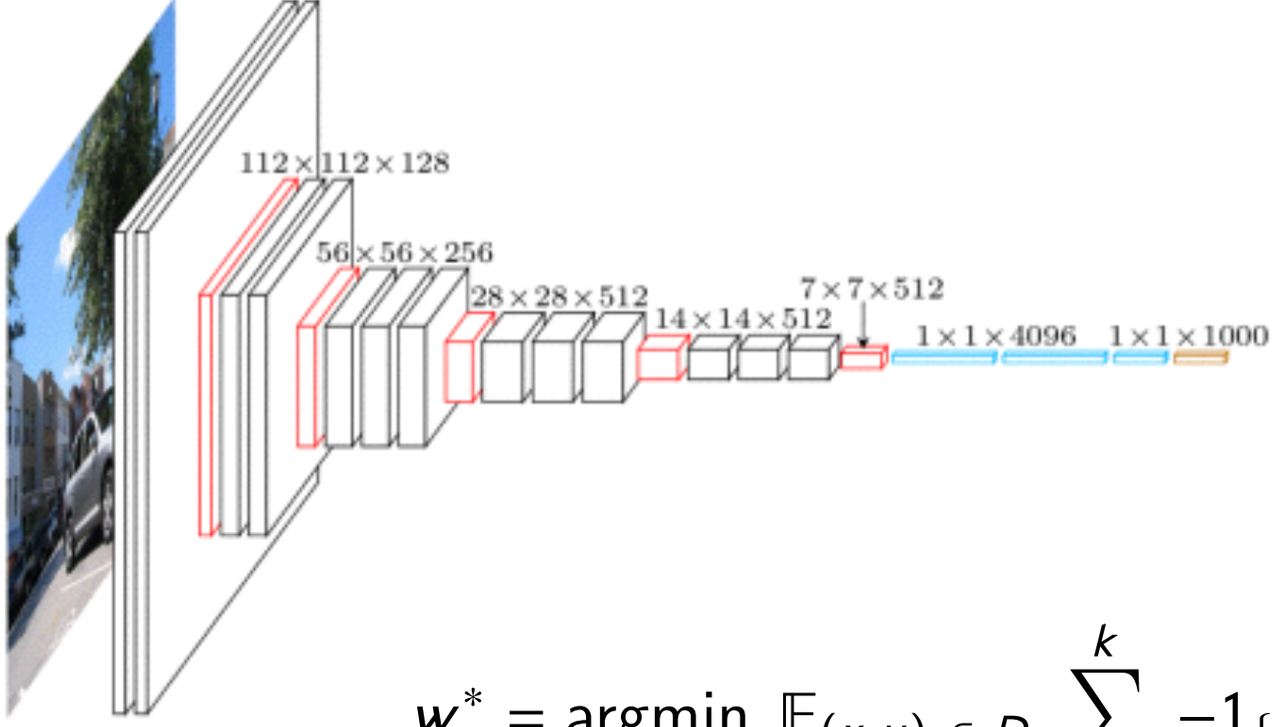
A picture of the energy landscape of deep neural networks

Pratik Chaudhari

December 15, 2017

UCLA VISION LAB



$$\widehat{y}(x; w) = \sigma(w^p\ \sigma(w^{p-1}\ (\ldots \sigma(w^1\ x))\ldots))$$

$$w^* = \operatorname*{argmin}_w \underbrace{\mathbb{E}_{(x,y)\ \in\ D} \sum_{i=1}^{k} -1_{\{y=i\}}\ \log \widehat{y}_i(x;\ w)}_{\triangleq\ f(w)}$$

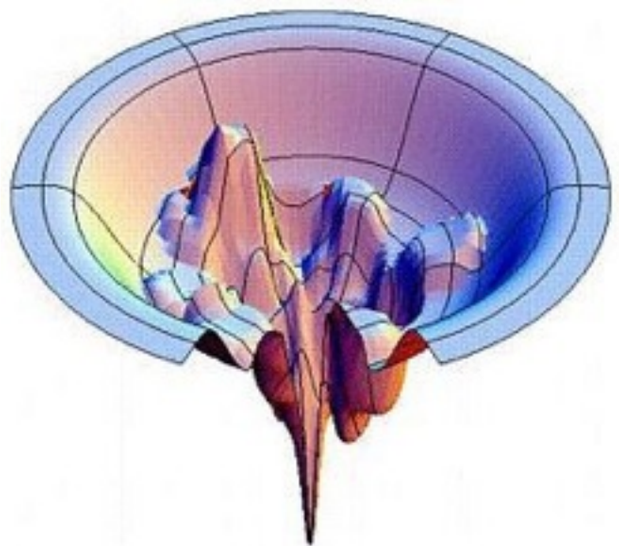
What is the shape?

(S)GD $w_{k+1} = w_k - \eta\ \nabla f_{\mathcal{U}}(w_k)$

Many, many variants

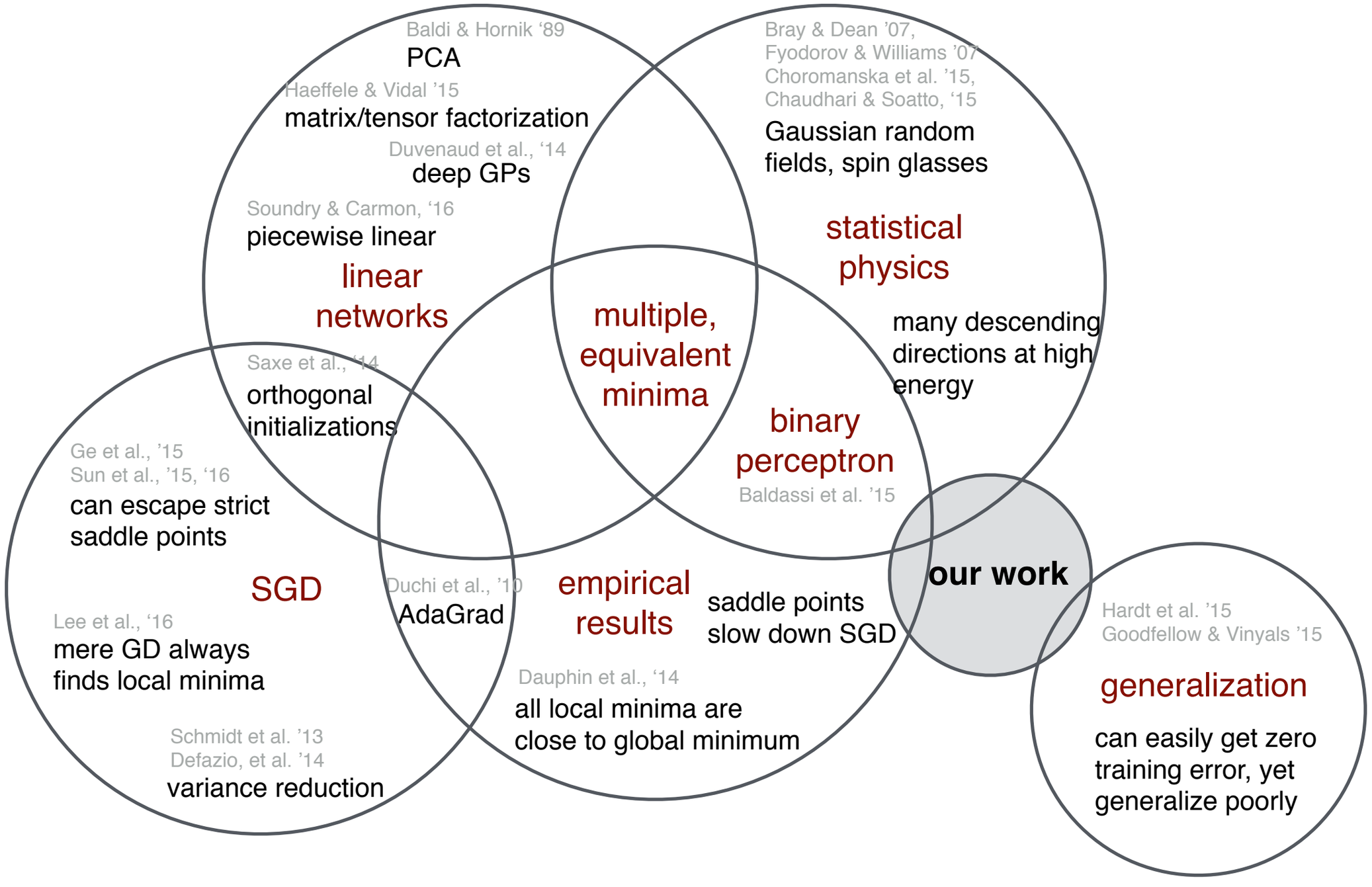
AdaGrad, SVRG, rmsprop, Adam, Eve,
APPA, Catalyst, Natasha, Katyusha…

"variance reduction"



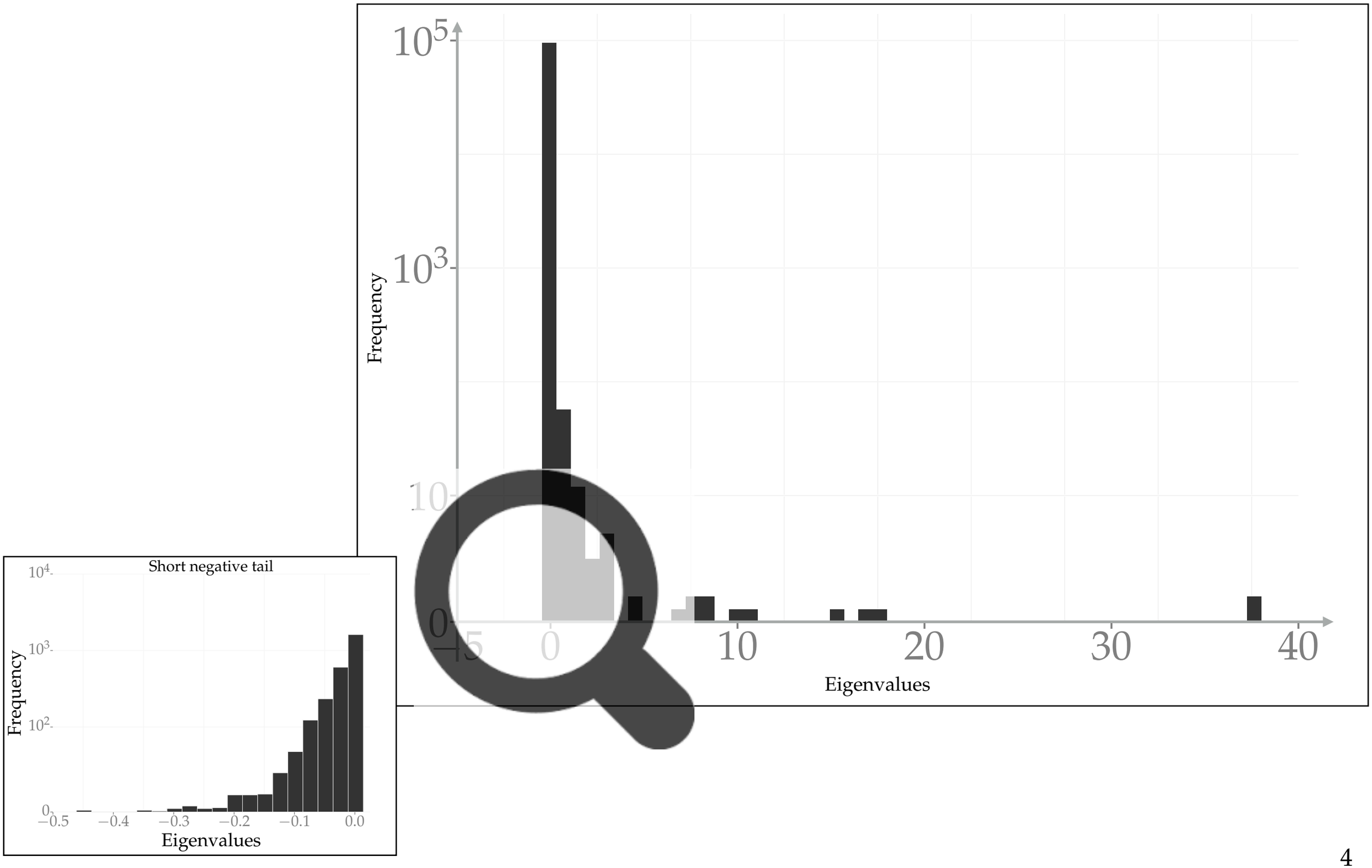
How should it inform the optimization?

What do we know about the energy landscape?



paradox between a “benign” energy landscape and delicate training algorithms

Motivation from the Hessian



How do we exploit this?

Magnify the energy landscape and smooth with a kernel

$$
\begin{aligned}
w^* &= \operatorname*{argmin}_{w} \; f(w) \\
&= \operatorname*{argmax}_{x} \; e^{-f(w)} \\
&\approx \operatorname*{argmin}_{w} \; -\log\left(G_\gamma \; * \; e^{-f(w)}\right)
\end{aligned}
$$

Gaussian kernel of variance γ

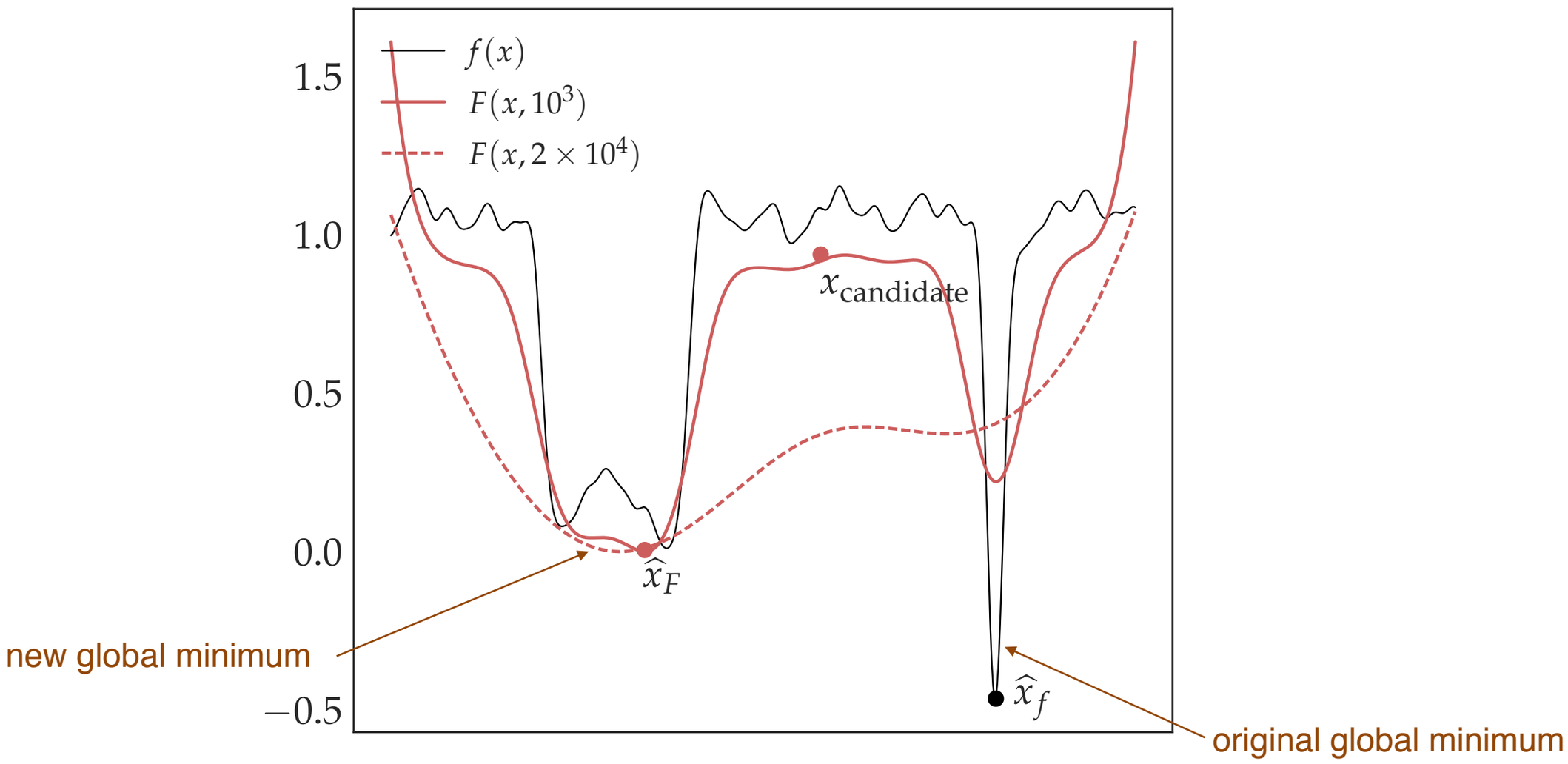
focuses on the neighborhood of w

A physics interpretation

Baldassi et al., ’15, ‘16

Our new loss is “local entropy”

$$f_\gamma(w) = -\log \int_{w'} \exp\left(-f(w') - \frac{1}{2\gamma}\,\|w - w'\|^2\right)\, dw'$$



Minimizing local entropy

- **Solve**

$$w^* = \underset{w}{\operatorname{argmin}}\ f_\gamma(w,\ \gamma)$$

- **Gradient of local entropy**

$$\nabla f_\gamma(w,\gamma) = \gamma^{-1}\left(w - \langle w' \rangle\right)$$

denotes an expectation over
a local Gibbs distribution

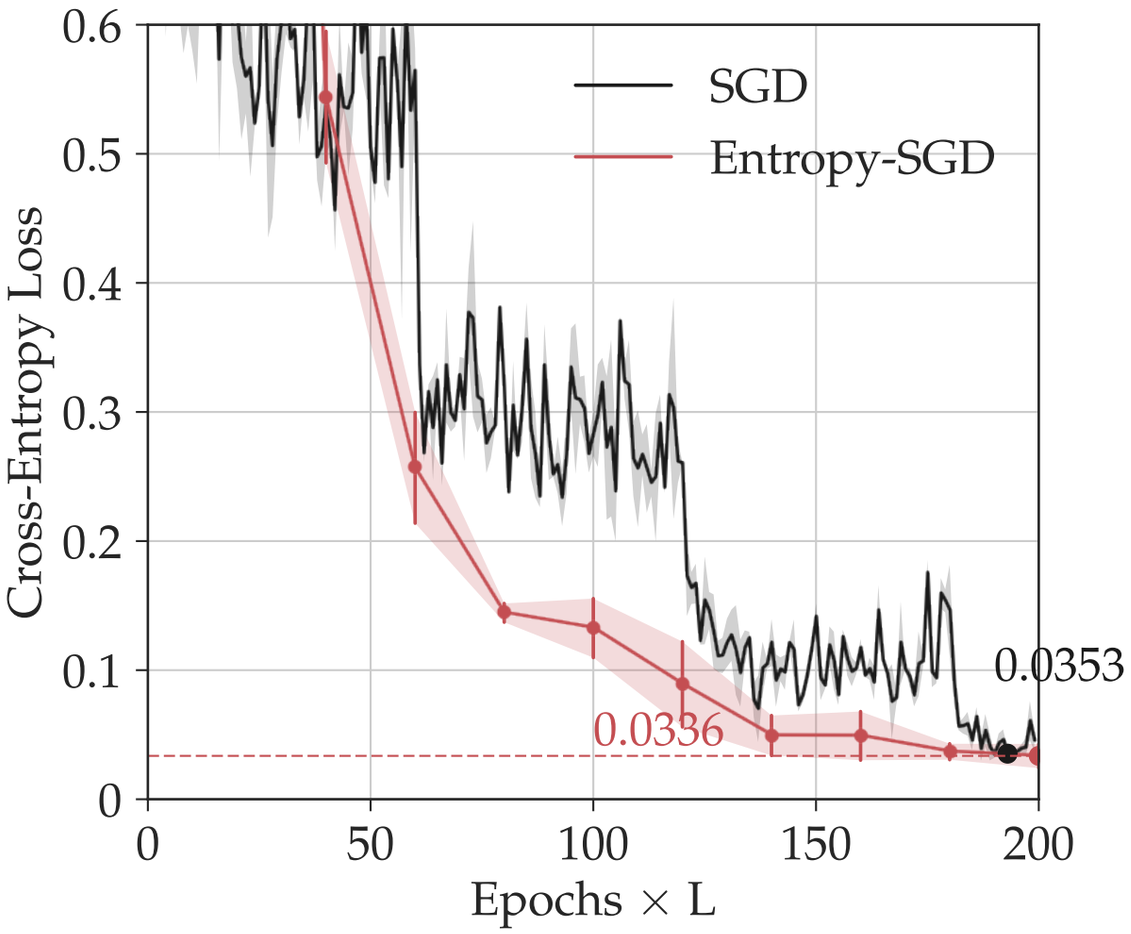
$$\langle w' \rangle = Z(w,\gamma)^{-1} \int_{w'} w' \ \exp\left(-f(w') - \frac{1}{2\gamma}\ \|w - w'\|^2\right)\ dw'$$

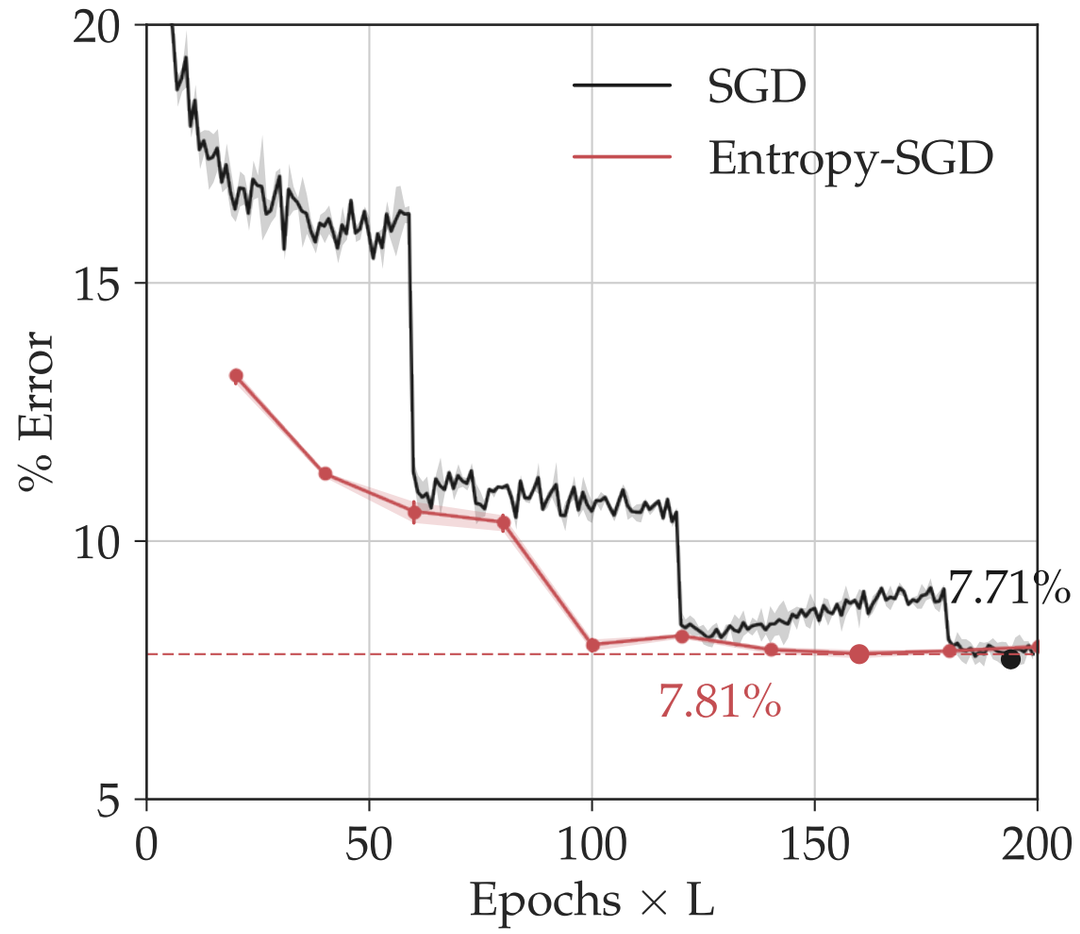
- **Estimate the gradient using MCMC**

  can be applied to general deep networks

Medium-scale CNN

- **All-CNN-BN on CIFAR-10**
- **Do not see much plateauing of training or validation loss**





x-axis ∝ wall-clock time

A PDE interpretation

- **Local entropy is the solution of a Hamilton-Jacobi equation**

$$u_t = -\frac{1}{2}\,|\nabla u|^2 + \frac{1}{2}\,\Delta\, u$$

$$u(w,0) = f(w)$$

original loss as the initial condition

$$f_\gamma(w) = u(w,\gamma)$$

- **Stochastic control interpretation**

$$dw = -\alpha(s)\,ds + dB(s), \quad t \le s \le T$$

$$w(t) = w.$$

$$C(w(\cdot),\ \alpha(\cdot)) = \mathbb{E}\left[f(w(T)) + \frac{1}{2}\int_t^T \|\alpha(s)\|^2\,ds\right]$$

quadratic penalty for greedy gradient descent

$$\alpha(w,t) = \nabla u(w,t)$$

$$u(w,t) = \min_{\alpha(\cdot)}\ C(w(\cdot),\ \alpha(\cdot))$$

New PDEs

- **Use the non-viscous HJ equation**

$$u_t = -\frac{1}{2}\,|\nabla u|^2 + \frac{1}{2}\,\Delta u$$

0

- **Hopf-Lax formula gives the solution**

"inf-convolution"
or Moreau envelope

$$u(w,t) = \inf_{w'}\left\{ f(w') + \frac{1}{2\gamma}\|w - w'\|^2 \right\}$$

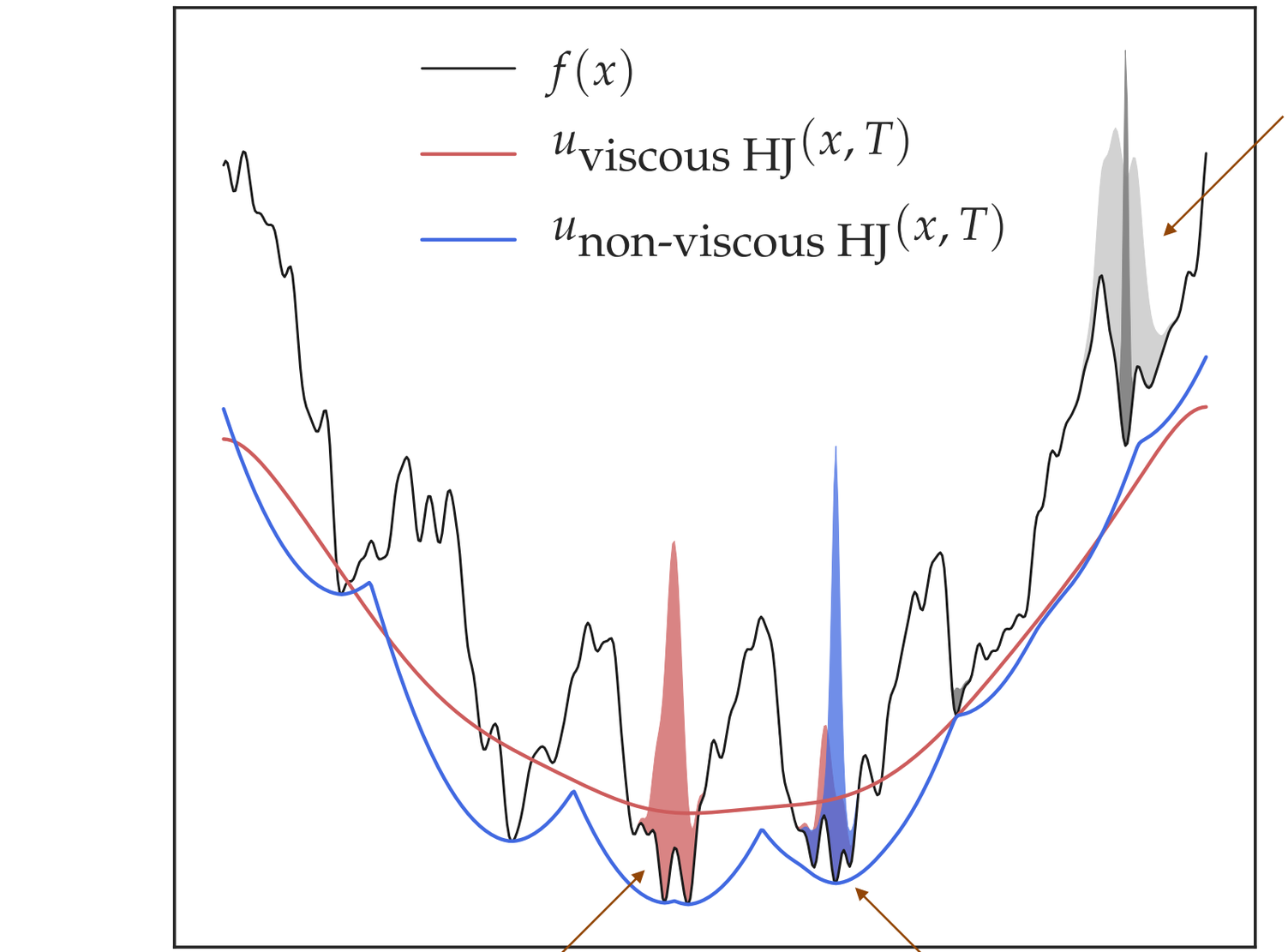
- **Simple formula for the gradient (proximal point iteration)**

$$\nabla u(w,t) = p^*$$

$$p^* = \nabla f(w - tp^*)$$

- **This has a few magical properties…**

Smoothing using PDEs



Distributed training algorithms

- **Elastic-SGD**

Zhang et. al., ’15

$$\underset{w,\ w'^1,\ldots,w'^p}{\operatorname{argmin}} \sum_{k=1}^{p} f(w'^k) + \frac{1}{2\gamma} \|w'^k - w\|^2$$

- **A continuous-time view of local entropy**

$$dw = -\gamma^{-1}(w - w')\, ds$$

fast variable

$$dw' = -\frac{1}{\epsilon}\left[\nabla f(w') + \frac{1}{\gamma}(w' - w)\right] ds + \frac{1}{\sqrt{\epsilon}}\, dB(s)$$

- **If *w’(s)* is very fast, *w(s)* only sees its average**

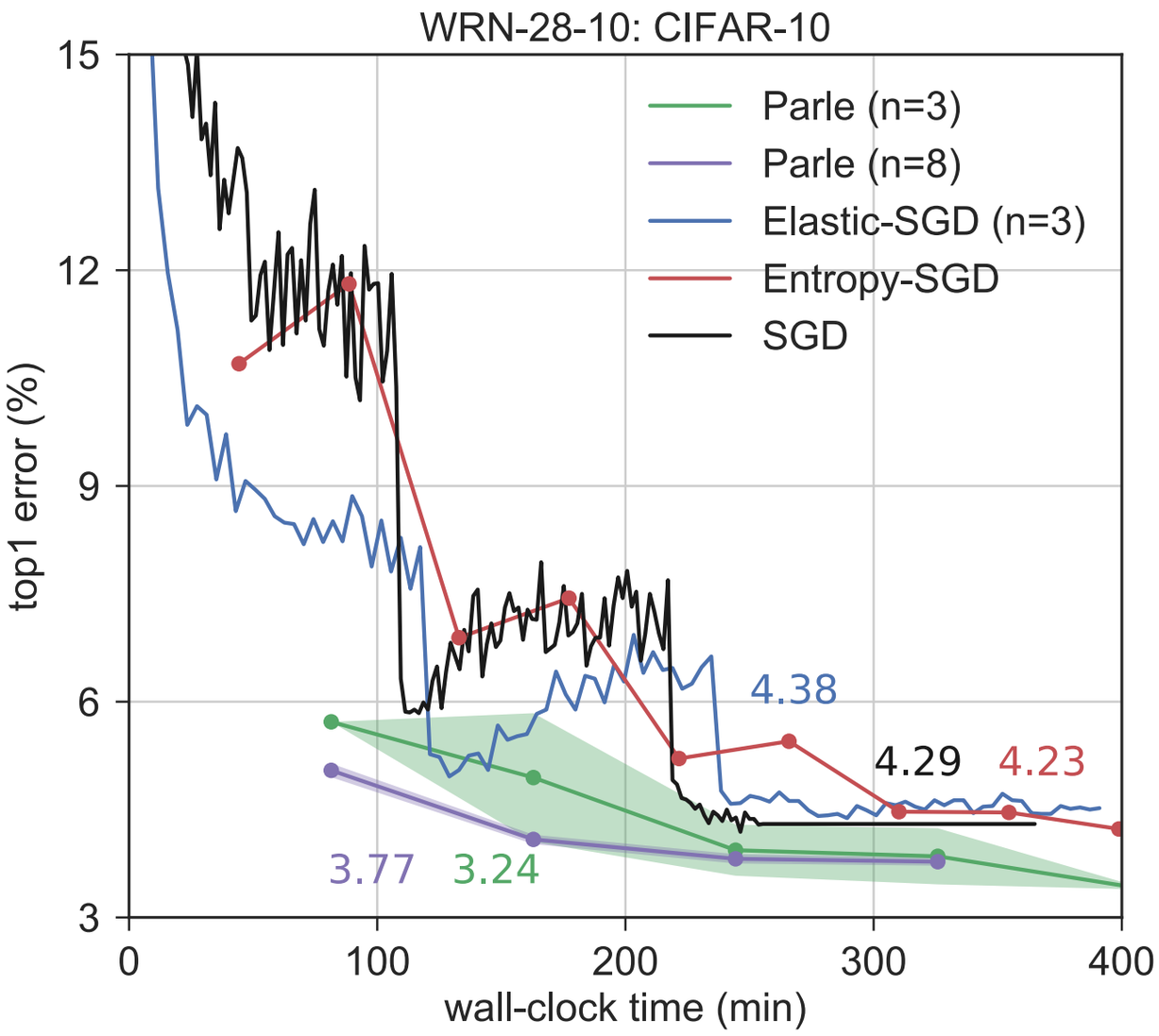
$$\nabla f_\gamma(w, \gamma) = \gamma^{-1}\left(w - \langle w' \rangle\right)$$

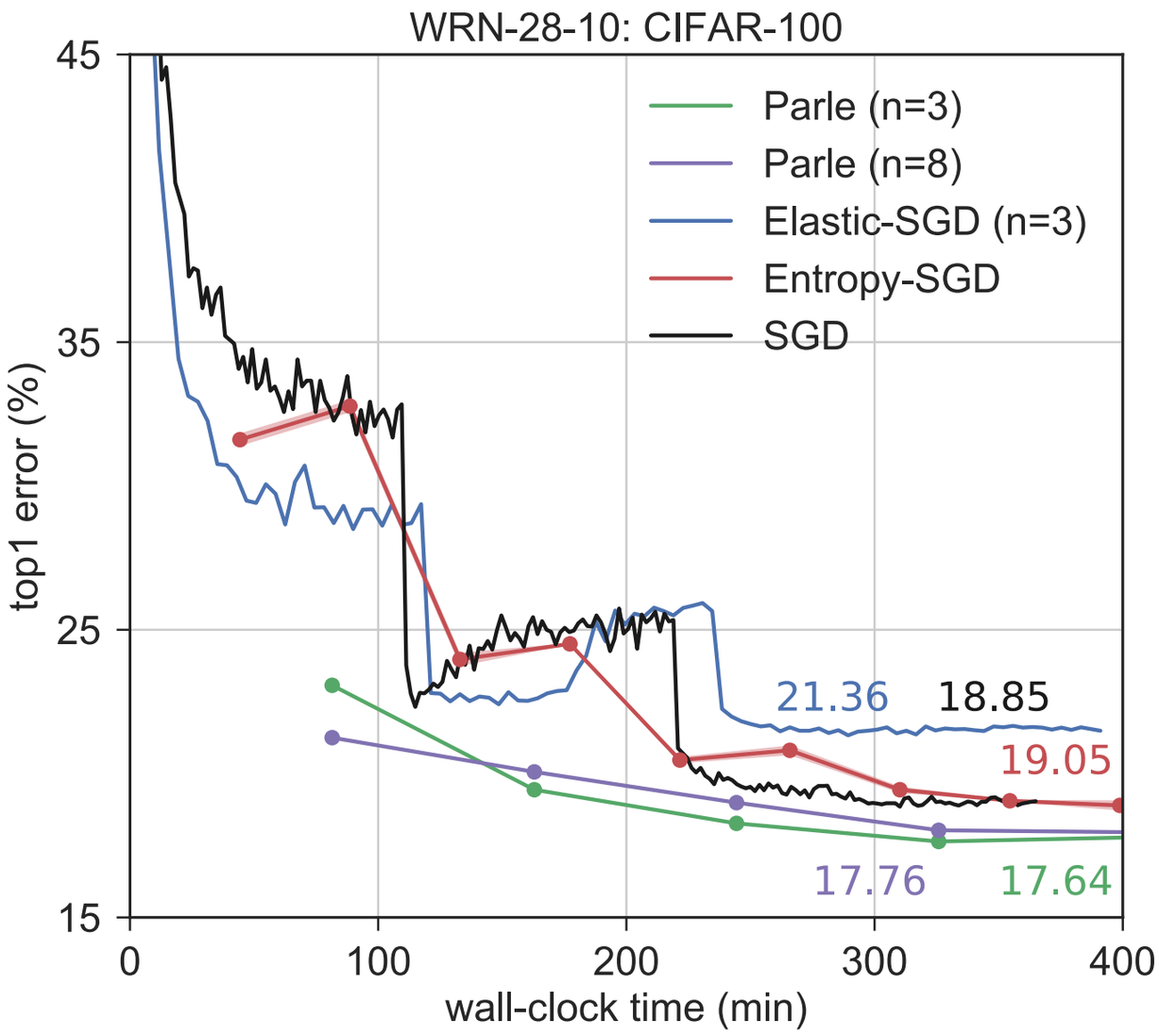
$$d\overline{w} = -\gamma^{-1}\left(\overline{w} - \langle w' \rangle\right) ds$$

“homogenized” dynamics as $\varepsilon \to 0$

$$\rho^{\infty}(w'; w) \propto \exp\left(-f(w') - \frac{1}{2\gamma}\|w' - w\|^2\right)$$

Wide-ResNet on CIFAR-10 / 100



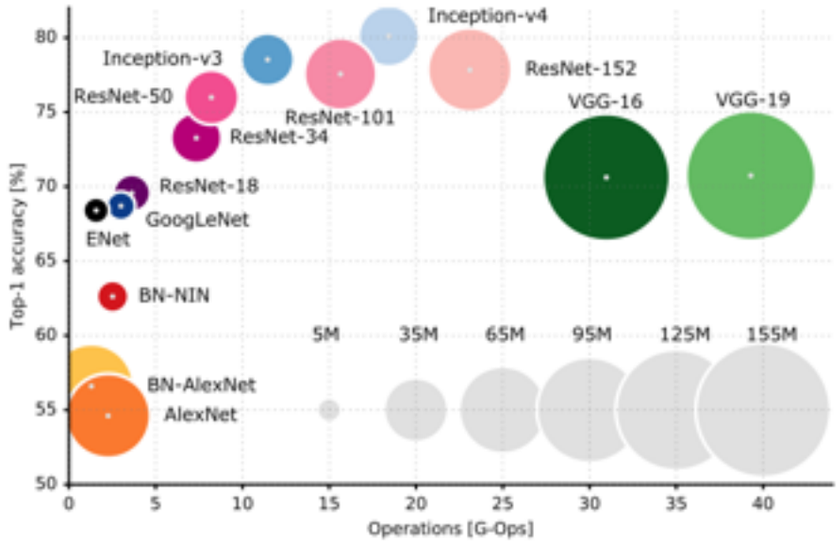


WTH is implicit regularization?

Many, many variants

AdaGrad, SVRG,
SAG, rmsprop,
Adam, Eve,
APPA, Catalyst,
Natasha, Katyusha…

Why is SGD so special?



- **Stochastic differential equation**

$$dw = -\nabla f(w)\underbrace{dt}_{\triangleq \eta} + \sqrt{2\beta^{-1} D(w)}\, dB(t) \qquad \beta^{-1} = \frac{\eta}{2\mathscr{C}}$$

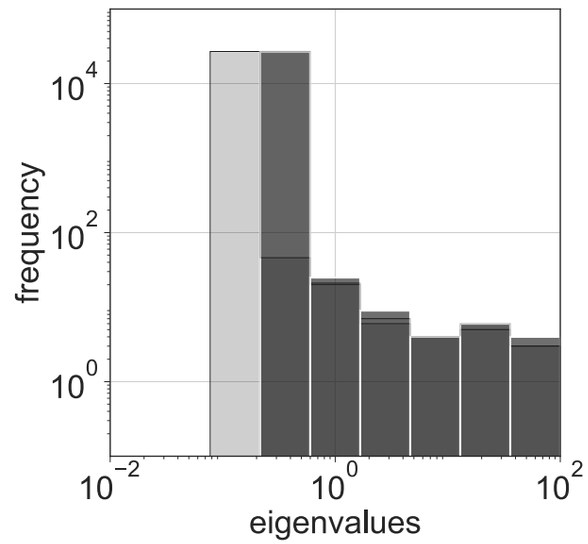
- **Fokker-Planck equation and optimal transportation**

$$\rho^* = \operatorname*{argmin}_{\rho} \int \Phi(w)\, \rho(w) dw \; + \beta^{-1} \int \log \rho \; d\rho.$$

Information bottleneck,
Bayesian inference,
large batch-sizes,
sampling techniques,
hyper-parameter choices,
neural architecture search,
....

SGD does not minimize $f(w)$, what is $\Phi(w)$?

- Deep networks induce highly non-isotropic noise



CIFAR-10

$\lambda(D) = 0.27 \pm 0.84$

$\text{rank}(D) = 0.34\%$

- Leads to deterministic, Hamiltonian dynamics in SGD

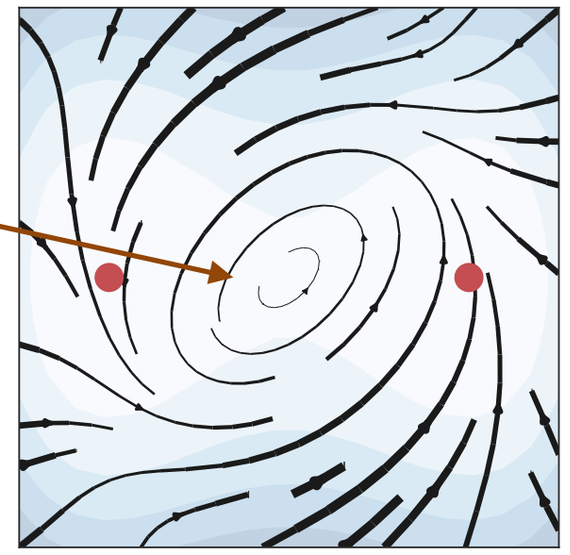
$$\dot{x} = j(x) \qquad \text{div}\, j(x) = 0$$

Most likely trajectories of SGD are closed loops

- Deep networks have out-of-equilibrium distribution

$$\rho^*(w) \not\propto e^{-\beta f(w)}$$

$$\propto e^{-\beta \Phi(w)}$$



Summary

- **Techniques from control and physics are interpretable, also lead to state-of-the-art algorithms**

  PDEs, stochastic control, stability of limit cycles, Fokker-Planck equations, continuous-time analysis…

- **Control has powerful tools to make inroads into understanding and improving deep networks**

  input-output stability, reinforcement learning & optimal control

- **Deep learning is powerful, and quite “easy” to get into**

  even the fundamentals are unknown and debated upon

Joint work with

Stefano Soatto

Guillaume Carlier, Adam Oberman, Stanley Osher

Yann LeCun, Anna Choromanska, Ameet Talwalkar

Carlo Baldassi, Christian Borgs, Jennifer Chayes, Riccardo Zecchina

www.pratikac.info

Thank You!