Entropic Gradient Descent Algorithms and Wide Flat Minima

Fabrizio Pittorino, Carlo Lucibello, Christoph Feinauer, Gabriele Perugini, Carlo Baldassi, Elizaveta Demyanenko, Riccardo Zecchina



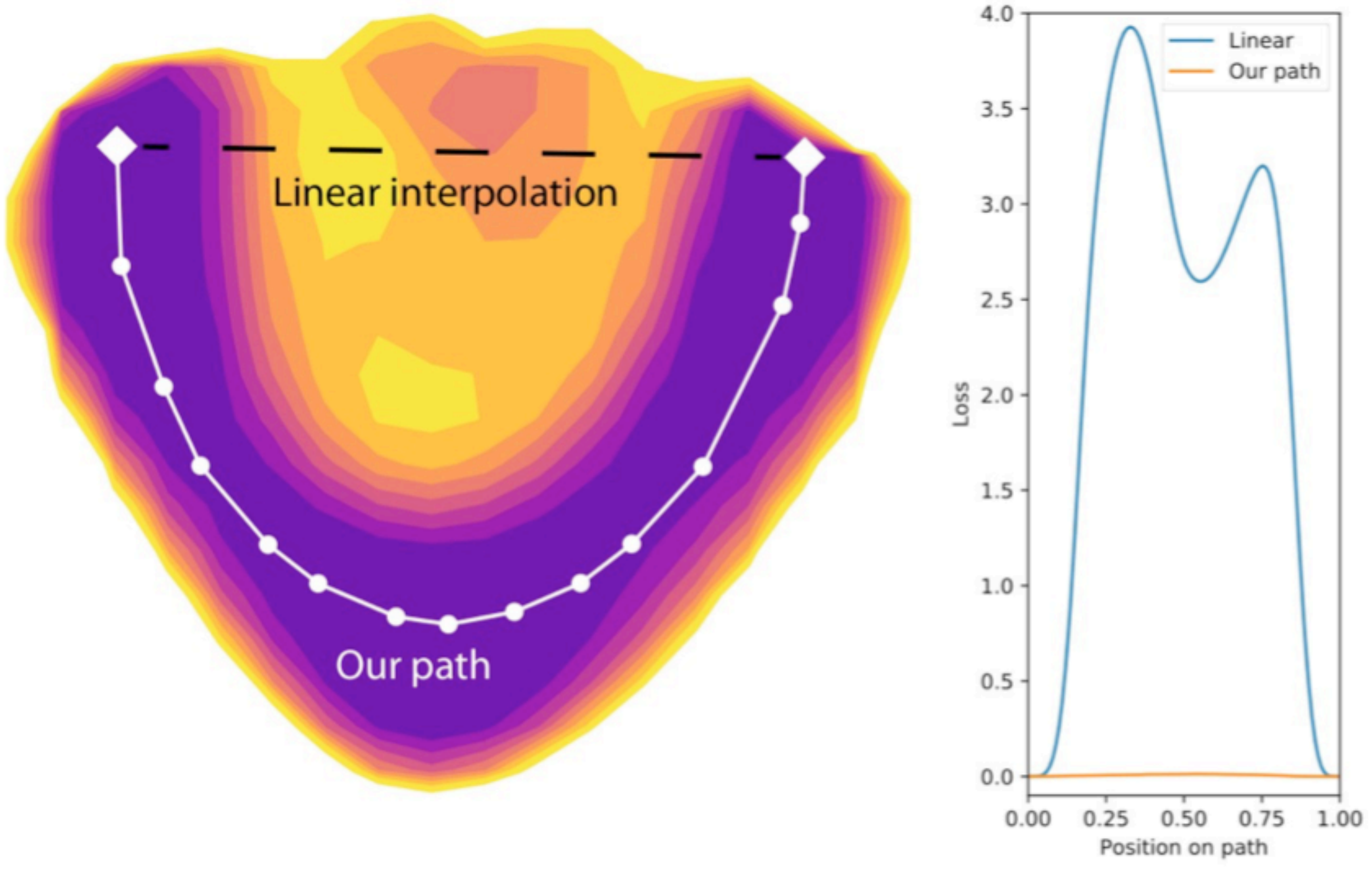

Artificial Intelligence Lab

Bocconi

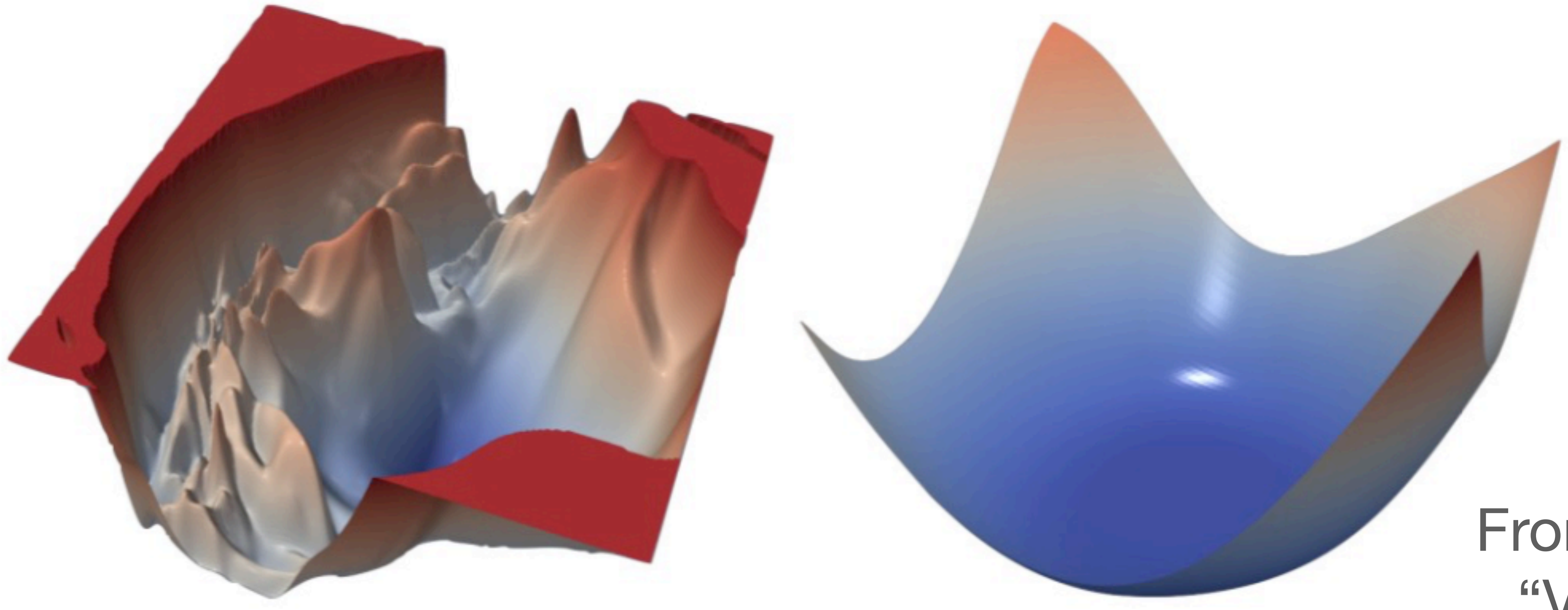


Complex Loss Landscapes in Neural Networks

- There seems to be a flat non-convex "bottom" connecting "accessible" minimizers.



From Draxler et al. '18, "Essentially no barriers..."

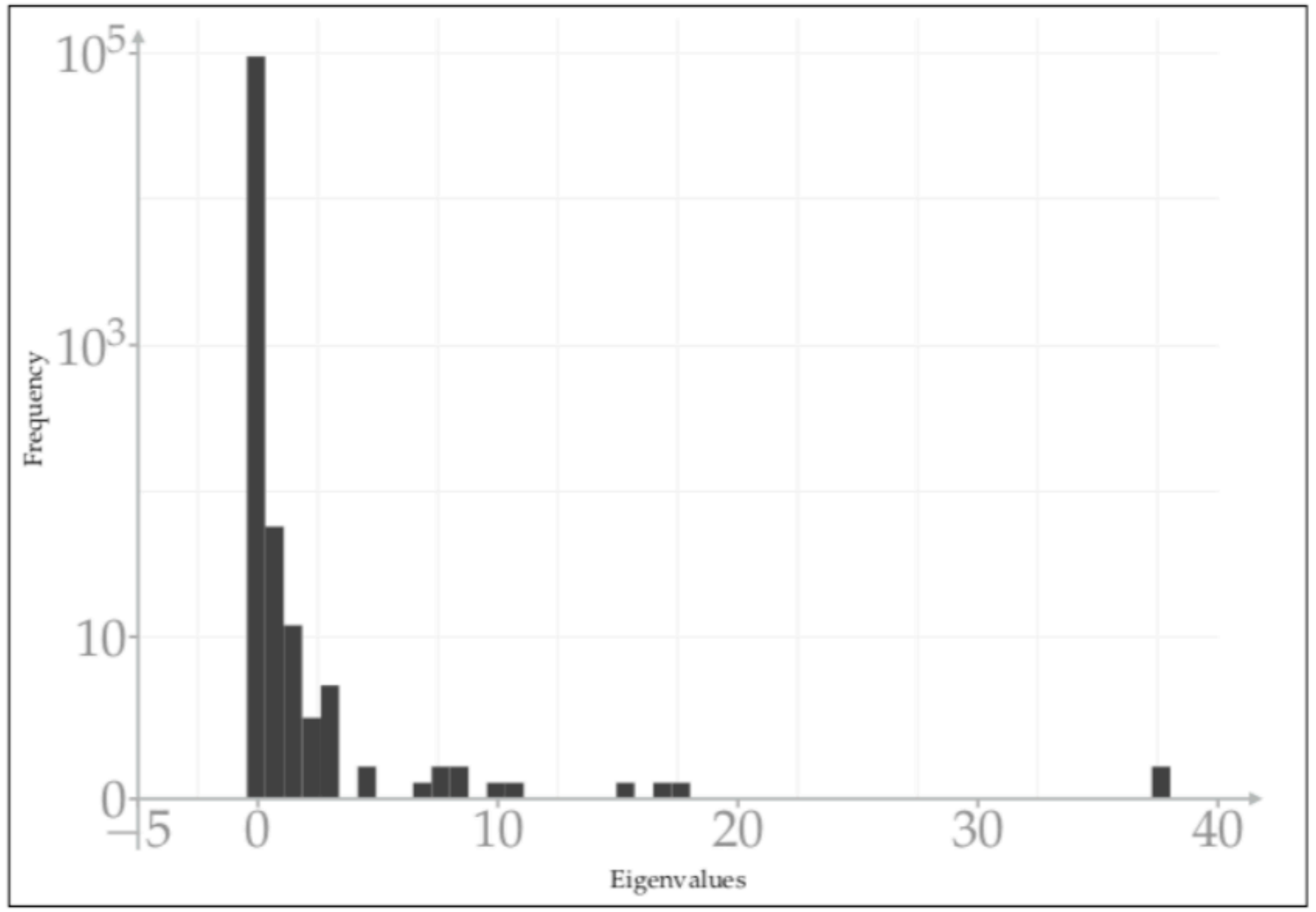


(a) ResNet-110, no skip connections

(b) DenseNet, 121 layers

From Li et al. '17, "Visualizing..."

- The spectra of the Hessian in a quasi-minimum. Many flat directions.



From Chaudhary et al '17 "EntropySGD..."

- Architectural choices (e.g. loss, activations, batch-norm, skip-connections) influence the roughness and the large-scale structure of the landscape

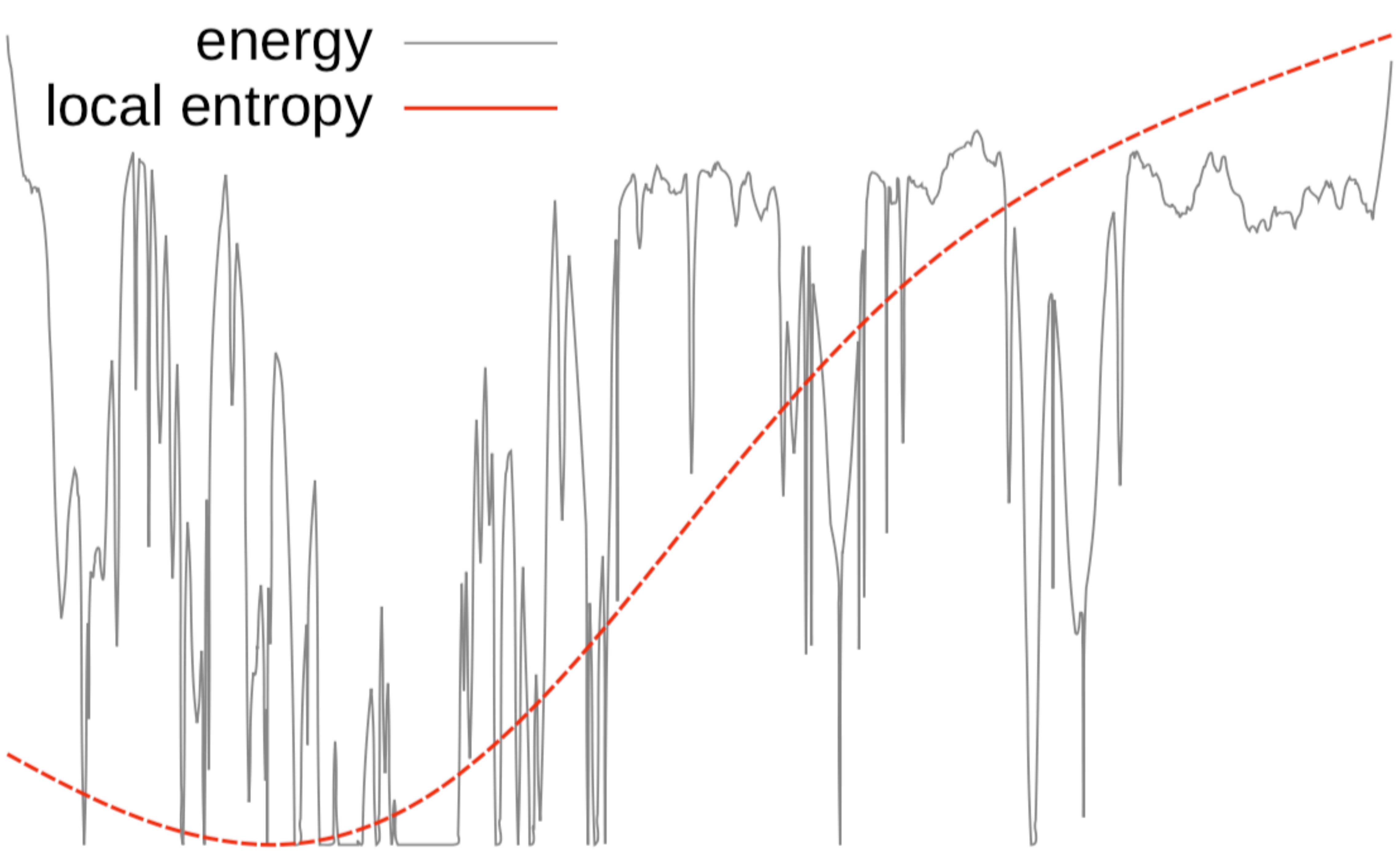
- SGD batch size anti-correlates with minima width and with generalization

Local Entropy and Local Energy

- How to tell apart good minima from bad minima?
- We conjecture that some geometrical properties of the traning loss landscape, and in particular the **flatness** of minima, correlates well with generalization
- We define the **local entropy loss** as a way to characterize flatness, and as an auxiliary loss:

$$\mathcal{L}_{\text{LE}}\left(w\right) = -\frac{1}{\beta}\log\int dw'\; e^{-\beta\mathcal{L}\left(w'\right)-\beta\gamma \mathrm{d}\left(w',w\right)}$$

Where the squared distance is: $\mathrm{d}(w',w) = \frac{1}{2}\sum_{i=1}^{N}\left(w_i' - w_i\right)^2$



- The **local energy** is a simple flatness measure:

$$\delta E_{\text{train}}(w,\sigma) = \mathbb{E}_z\, E_{\text{train}}(w + \sigma z \odot w) - E_{\text{train}}(w)$$

Where the noise is: $z \sim \mathcal{N}(0, I_N)$

Entropic Algorithms

- Local Entropy hard to compute,
- but the gradient: $\nabla \mathcal{L}_{\mathrm{LE}}(w) = \gamma \left(w - \langle w' \rangle \right)$
- can be approximated by Stochastic Gradient Langevin Dynamics. The corresponding algorithm is called Entropy-SGD [1]

Algorithm 1: Entropy-SGD (eSGD)

Input : w
Hyper-parameters : $L, \eta, \gamma, \eta', \epsilon, \alpha$

for $t = 1, 2, \ldots$ **do**
- $w', \mu \leftarrow w$
- **for** $l = 1, \ldots, L$ **do**
  - $\Xi \leftarrow$ sample minibatch
  - $dw' \leftarrow \nabla \mathcal{L}\left(w'; \Xi\right) + \gamma \left(w' - w\right)$
  - $w' \leftarrow w' - \eta' dw' + \sqrt{\eta'}\, \epsilon \mathcal{N}(0, I)$
  - $\mu \leftarrow \alpha \mu + (1 - \alpha)\, w'$
- $w \leftarrow w - \eta \left(w - \mu\right)$

Algorithm 2: Replicated-SGD (rSGD)

Input : $\{w^a\}$
Hyper-parameters : y, η, γ, K

for $t = 1, 2, \ldots$ **do**
- $\bar{w} \leftarrow \frac{1}{y} \sum_{a=1}^{y} w^a$
- **for** $a = 1, \ldots, y$ **do**
  - $\Xi \leftarrow$ sample minibatch
  - $dw^a \leftarrow \nabla \mathcal{L}\left(w^a; \Xi\right)$
  - **if** $t = 0 \bmod K$ **then**
    - $dw^a \leftarrow dw^a + K\gamma \left(w^a - \bar{w}\right)$
  - $w^a \leftarrow w^a - \eta\, dw^a$

- Another class of entropic algorithms can be derived starting from

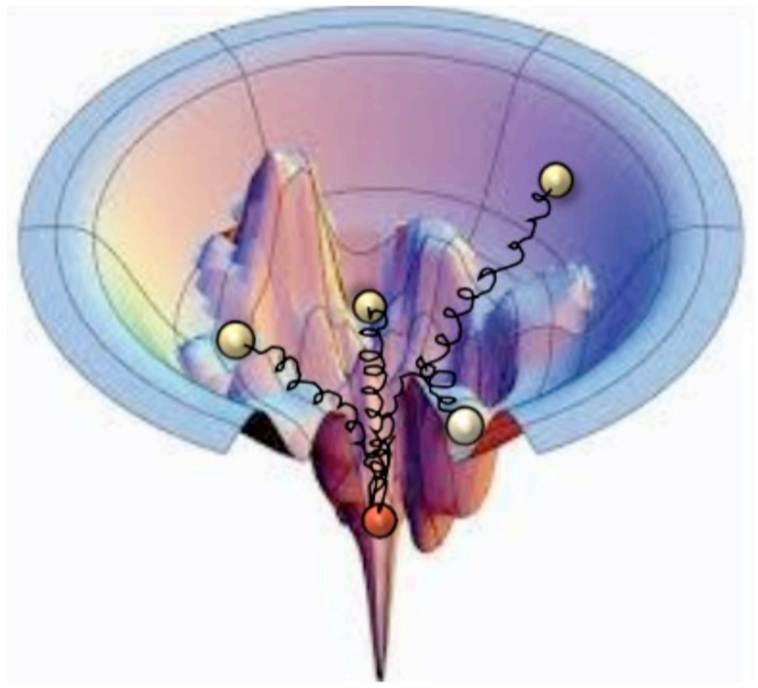
$$p(w) \propto e^{-\beta y \mathcal{L}_{\mathrm{LE}}(w)}$$

- For y integer, one can use the local entropy definition to obtain the statistical measure of a system with y+1 replicas, then integrate out the original one and obtain:

$$p(\{w^a\}_{a=1}^{y}) \propto e^{-\beta \mathcal{L}_{\mathrm{R}}(\{w^a\})}$$

- Where $\mathcal{L}_R(\{w^a\}_a) = \sum_{a=1}^{y} \mathcal{L}(w^a) + \gamma \sum_{a=1}^{y} \mathrm{d}\left(w^a, \bar{w}\right)$

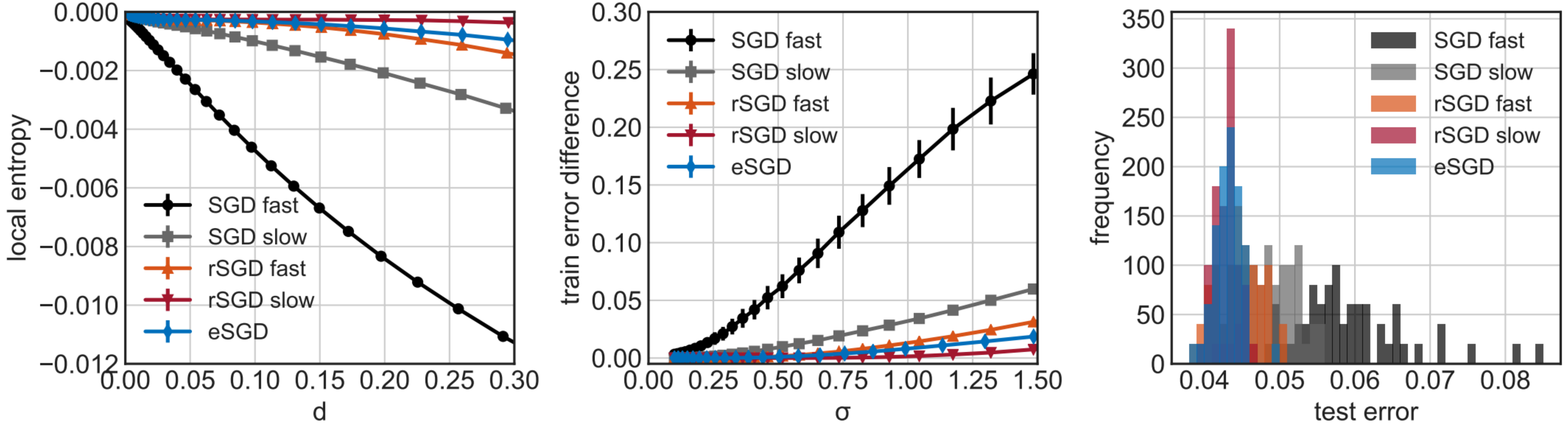
- with $\bar{w} = \frac{1}{y} \sum_a w^a$. Now one can perform SGD on the replicated loss.



Shallow networks: estimation by Belief Propagation

- Shallow network performing a binary classification task on Fashion-MNIST
- We use Belief Propagation to estimate local entropy
- Local entropy and local energy correlate with each other and with generalisation

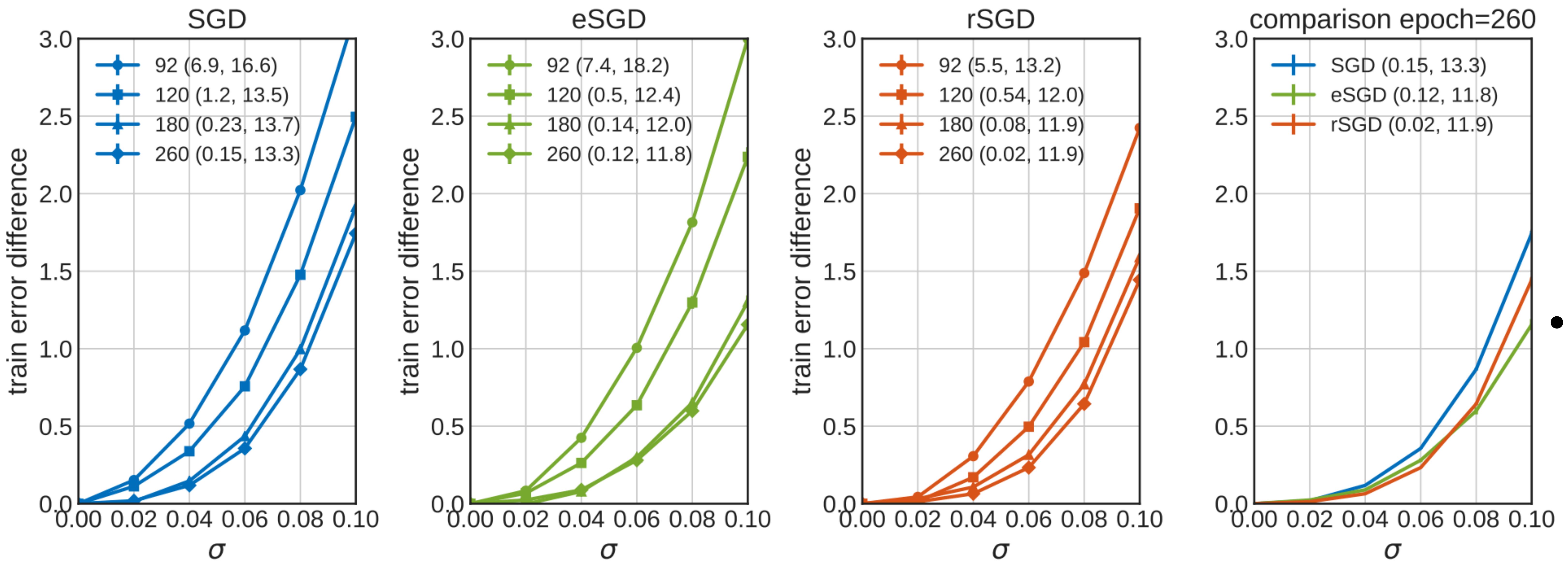
$$\hat{\sigma}(w,x) = \mathrm{sign}\left[\frac{1}{\sqrt{K}}\sum_{k=1}^{K}\mathrm{sign}\left(\frac{1}{\sqrt{N}}\sum_{i=1}^{N}w_{ki}x_i\right)\right]$$



Deep Networks: flatness and generalisation

Dataset	Model	Baseline	rSGD	eSGD	rSGD×y
CIFAR-10	SmallConvNet	16.5 ± 0.2	15.6 ± 0.3	14.7 ± 0.3	14.9 ± 0.2
	ResNet-18	13.1 ± 0.3	12.4 ± 0.3	12.1 ± 0.3	11.8 ± 0.1
	ResNet-110	6.4 ± 0.1	6.2 ± 0.2	6.2 ± 0.1	5.3 ± 0.1
	PyramidNet+ShakeDrop	2.1 ± 0.2	2.2 ± 0.1		1.8
CIFAR-100	PyramidNet+ShakeDrop	13.8 ± 0.1	13.5 ± 0.1		12.7
	EfficientNet-B0	20.5	20.6	20.1 ± 0.2	19.5
Tiny ImageNet	ResNet-50	45.2 ± 1.2	41.5 ± 0.3	41.7 ± 1	39.2 ± 0.3
	DenseNet-121	41.4 ± 0.3	39.8 ± 0.2	38.6 ± 0.4	38.9 ± 0.3

- We want to verify that our entropic algorithm effectively find flatter minima.



- Local Entropy is expensive to compute, we compute the cheap Local Energy:

$$\delta\epsilon(w) = \mathbb{E}_z\, \epsilon(w(1+\sigma z)) - \epsilon(w)$$

- Confirming that entropic algorithm find flatter minima and generalize better

Conclusions

- Local entropy and local energy correlate with each other and with generalisation
- Detailed comparison on shallow networks (semi-analytical study)
- For deep networks, we showed entropic algorithms outperform standard ones, having enhanced generalisation and flatness