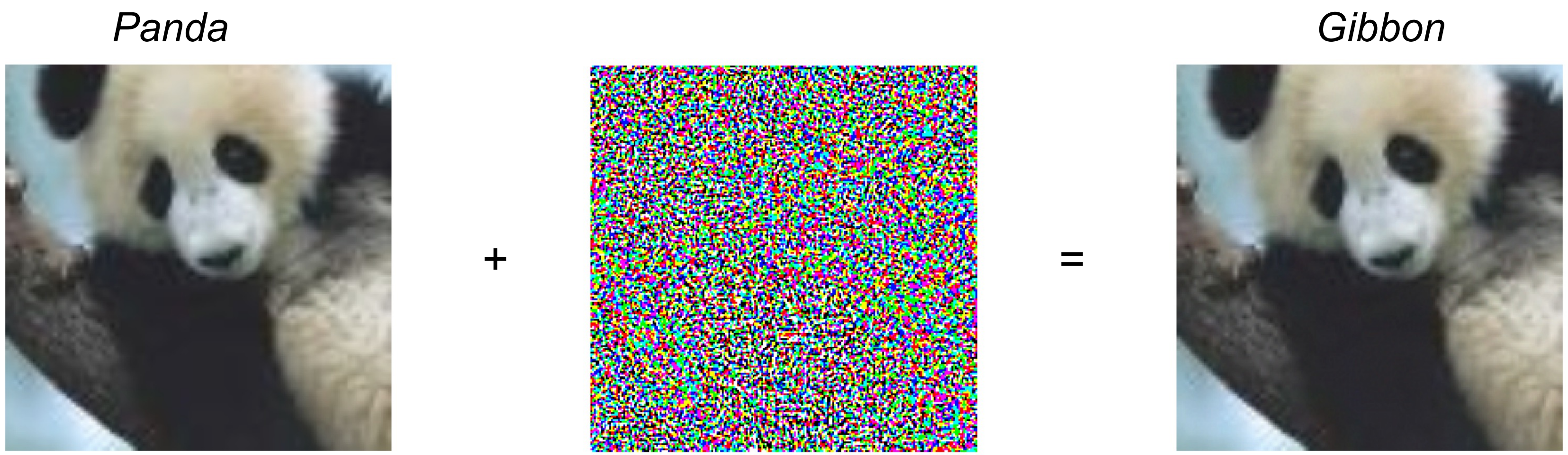
The Uncanny Valley: Exploring Adversarial robustness from a flatness perspective

Walter, N. P., Adilova, L., Vreeken, J., & Kamp, M. (2024). **The Uncanny Valley: Exploring Adversarial Robustness from a Flatness Perspective**.

Adversarial Examples



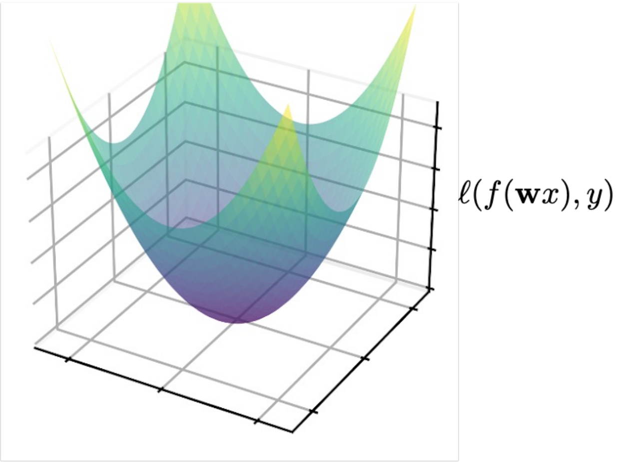
A lot of research, but still an open problem!

Can analyzing flatness help?

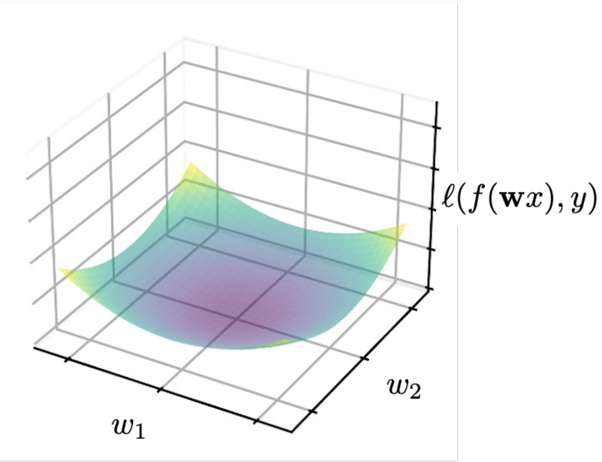
Explaining And Harnessing Adversarial Examples, Goodfellow et. al., ICLR 2015

What is flatness?

"Large region in weight space with the property that each weight vector from that region leads to similar small error"

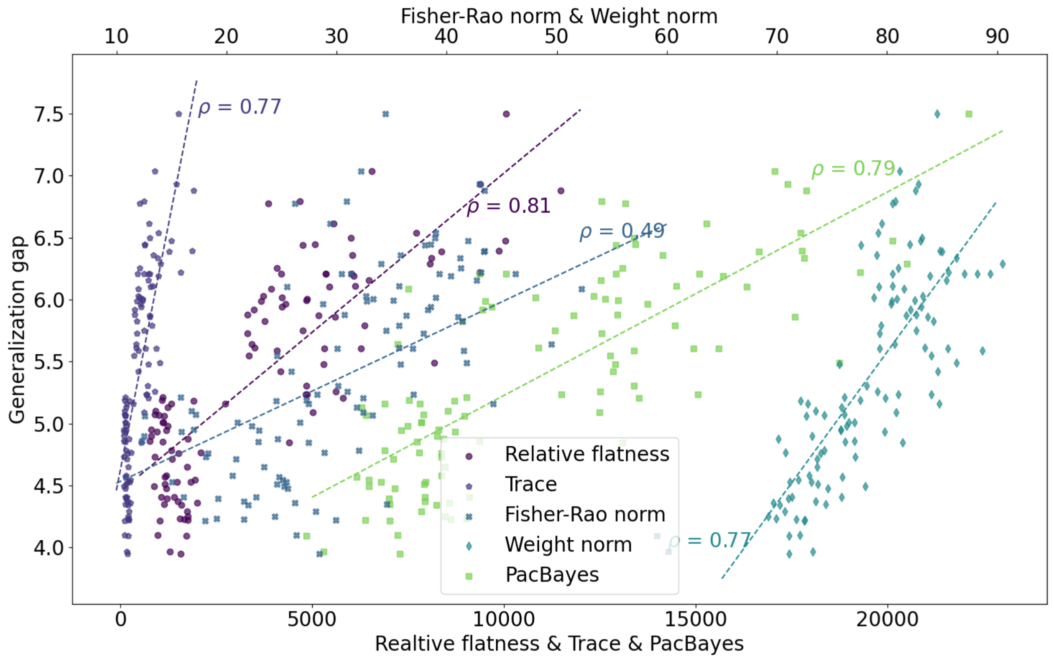


Sharp



Flat

Flatness and Generalization



Provably and Empirically shown to correlate with generalization!

Generalization

What is generalization?

→ Difference between error on the overall distribution minus the empirical distribution

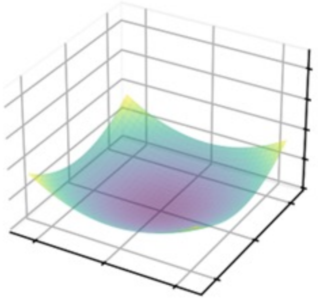
$$\mathcal{E}_{gen}(f,S) = \mathcal{E}_{\mathcal{D}}(f) - \mathcal{E}_{emp}(f,S)$$

ε-representativeness feature robustness

$\mathcal{E}_{gen}(f,S) =$ global property + local property

best and worst-case bounds

flatness



The Loss Surface and Generalization, Petzka & **Kamp** & **Adilova** et.al. , Neurips, 2021.

Measuring Flatness

Empirical

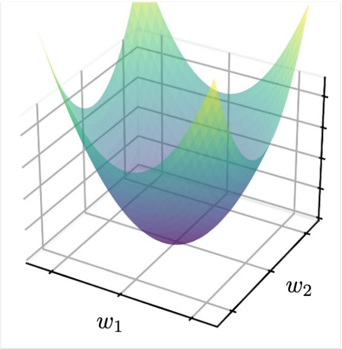
Avg-case:

$$S^{\rho}_{avg}(\boldsymbol{w}, \boldsymbol{c}) \triangleq \mathbb{E}_{\substack{\mathcal{S}\sim P_m \\ \boldsymbol{\delta}\sim\mathcal{N}(0,\rho^2 diag(\boldsymbol{c}^2))}} L_{\mathcal{S}}(\boldsymbol{w}+\boldsymbol{\delta}) - L_{\mathcal{S}}(\boldsymbol{w})$$

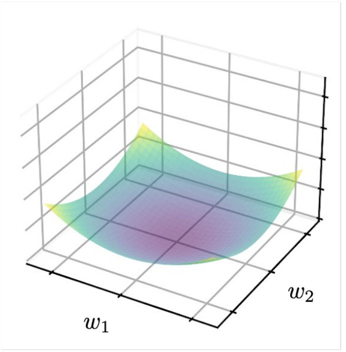
Worst-case:

$$S^{\rho}_{max}(\boldsymbol{w}, \boldsymbol{c}) \triangleq \mathbb{E}_{\mathcal{S}\sim P_m} \max_{\|\boldsymbol{\delta}\odot\boldsymbol{c}^{-1}\|_p\leq\rho} L_{\mathcal{S}}(\boldsymbol{w}+\boldsymbol{\delta}) - L_{\mathcal{S}}(\boldsymbol{w})$$

Sharp



Flat



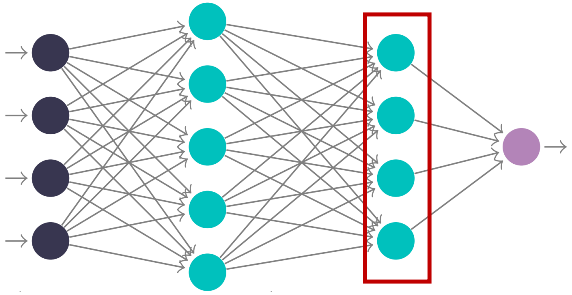
Analytical

Relative flatness:

$$K^{\phi}_{Tr} = \|\mathbf{w}\|_2 Tr(H)$$

For the penultimate layer and CE-loss

$$H = diag(\hat{y}) - \hat{y}\hat{y}^{\top} \otimes \phi\phi^{\top}$$
with
$$\hat{y} = f(x) = \psi(\phi(x)) \text{ and } \phi = \phi(x)$$



Connection to adversarial robustness

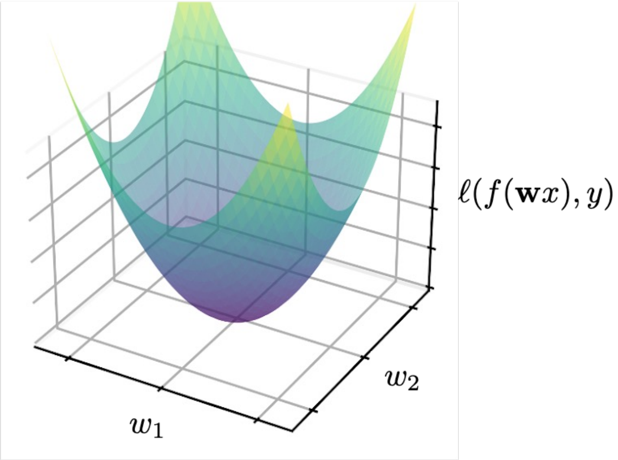
Main idea: Changes in the input

↓

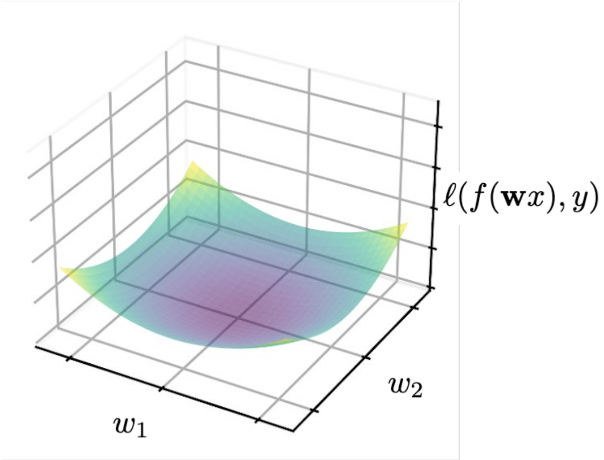
Changes in the weights

$$
\begin{aligned}
\ell(f(\mathbf{w}(x+\delta x)), y) &= \ell(f(\mathbf{w}x + \mathbf{w}\delta x)), y) \\
&= \ell(f(\mathbf{w} + \mathbf{w}\delta)x), y) \\
&= \ell(f(\hat{\mathbf{w}}x), y) \\
&\approx \ell(f(\mathbf{w}x), y)
\end{aligned}
$$

→ Loss is similar; hence , similar prediction



Sharp



Flat

Connection to adversarial robustness II

Definition 1 (Szegedy et al. (2014); Papernot et al. (2016) and Carlini & Wagner (2017b)). *Let $f : \mathbb{R}^m \to \{1, \dots, k\}$ be a classifier, $x \in [0,1]^m$, and $l \in [k]$ with $l \neq f(x)$ a target class. Then for every*

$$r^* = \arg\min_{r \in \mathbb{R}^m} \|r\|_2 \text{ s.t. } f(x+r) = l \text{ and } x + r \in [0,1]^m$$

the perturbed sample $\xi = x + r^*$ is called an ***adversarial example***.

Explicitly incorporate distance measure

Definition 2. *Let $\mathcal{D}$ be a distribution over an input space $\mathcal{X}$ and a label space $\mathcal{Y}$ with corresponding probability density function $P(X,Y) = P(Y \mid X)P(X)$. Let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ be a loss function, $f \in \mathcal{F}$ a model, and $(x,y) \in \mathcal{X} \times \mathcal{Y}$ be an example drawn according to $\mathcal{D}$. Given a distance function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ over $\mathcal{X}$ and two thresholds $\epsilon, \delta \geq 0$, we call $\xi \in \mathcal{X}$ an* ***adversarial example*** *for x if $d(x, \xi) \leq \delta$ and*

$$\mathbb{E}_{y_\xi \sim P(Y|X=\xi)} \left[\ell(f(\xi), y_\xi)\right] - \ell(f(x), y) > \epsilon \ .$$

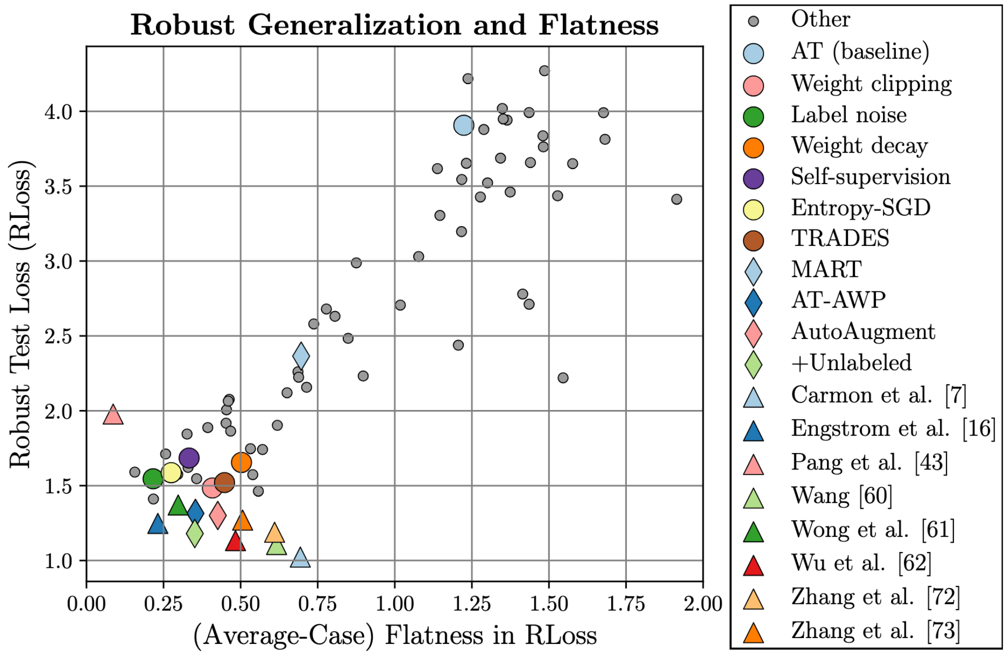
This is a (ϵ, δ)-criterion i.e. ***smoothes!***

Simplifying neural nets by discovering flat minima, Hochreiter and Schmidhuber, NIPS, 1995.

Empirical Evidence

Adversarial robustness correlates with flatness.

Loss on adversarial examples



Various adversarially trained models

Sample weights in a neighborhood and eval flatness

Our results

Setup

- Take model e.g. WideResnet18
- Train on CIFAR10
- Attack with an iterative attack e.g. PGD with delta=8/255 and 10 iteration
  → sufficient for ~0% Acc
- We do not stop the attack

Evaluate flatness measure on each adversarial distribution S_i^{adv}

Path adversarial image

$i = 0$

$i = 1$

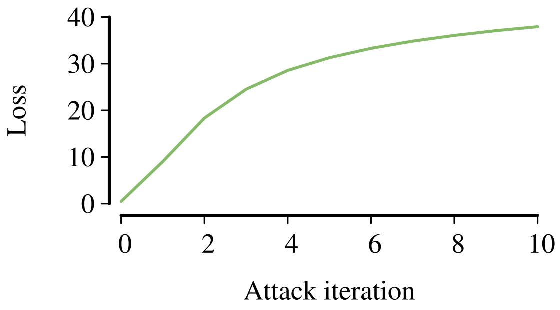
⋮

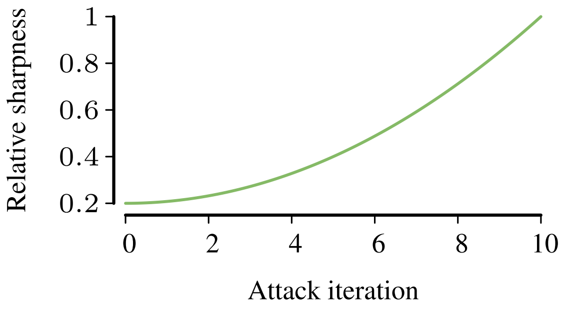
$i = 10$

What do you expect?

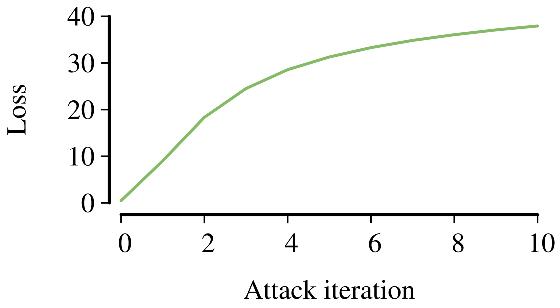
Results

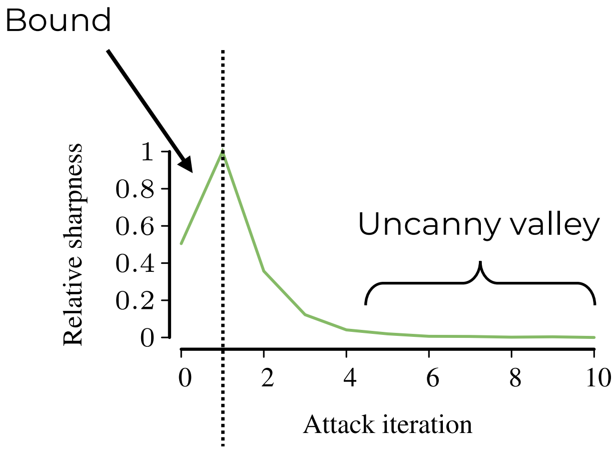
The curves we deserved





The curves we got





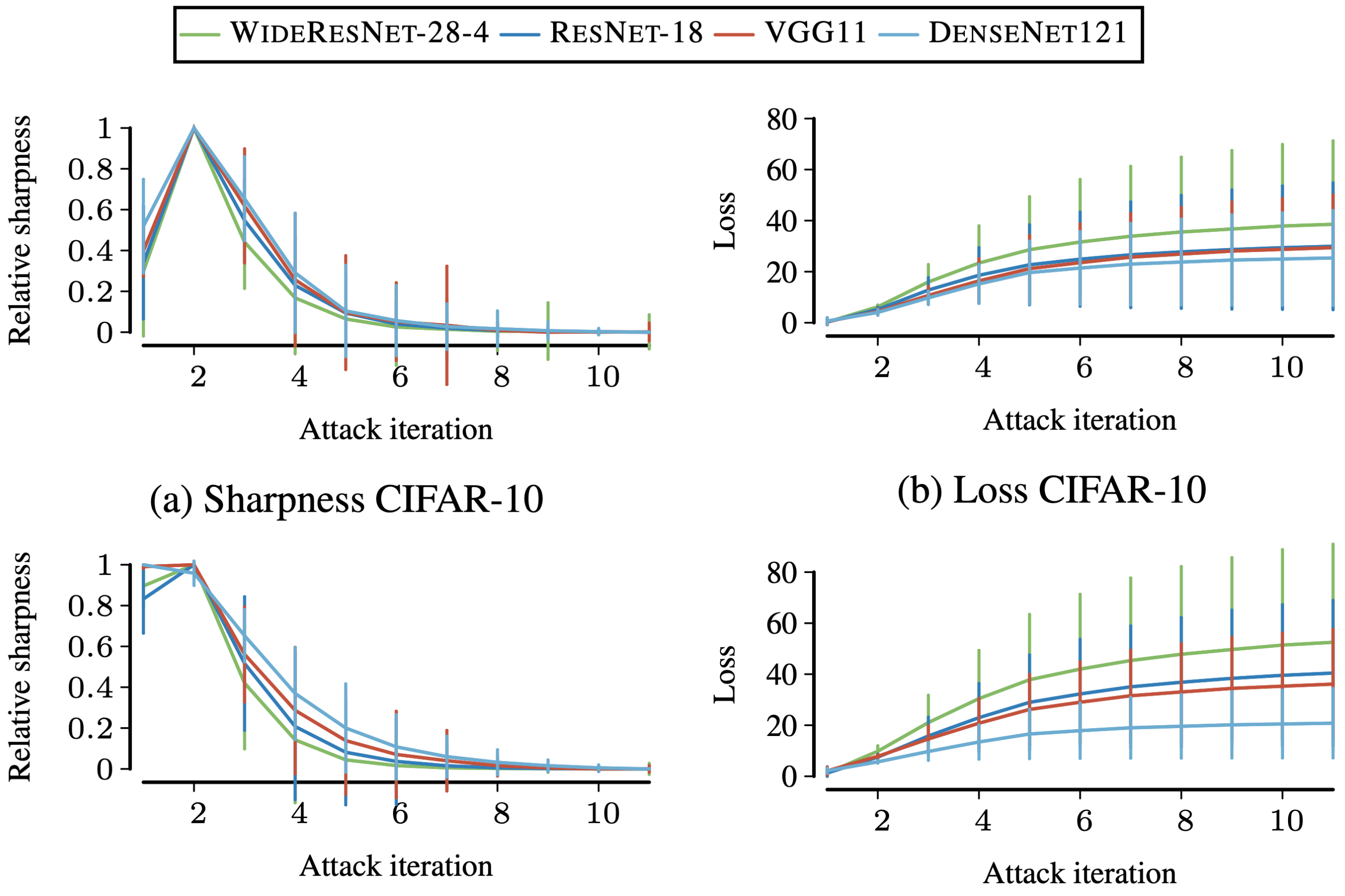
Bounding adversarial robustness

For a given dataset $\mathcal{S}$ we can guarantee δ-adversarial robustness with

$$\delta \propto \frac{\epsilon^{\frac{1}{3}}}{(L^3 k_{Tr}(\mathbf{w}))^{\frac{1}{3}}} + \frac{rkmL^2}{k_{Tr}(\mathbf{w})}$$

- Relative flatness K_{Tr}
- Lipschitz constant L
- Dimension of representation m, i.e., $\phi(x) \in \mathbb{R}^m$
- Number of output neurons k
- Lower bound on representation norm r, i.e., $\forall x : \|\phi(x)\| \geq r$
- Loss difference between clean and adversarial example ϵ

The Uncanny Valley is everywhere



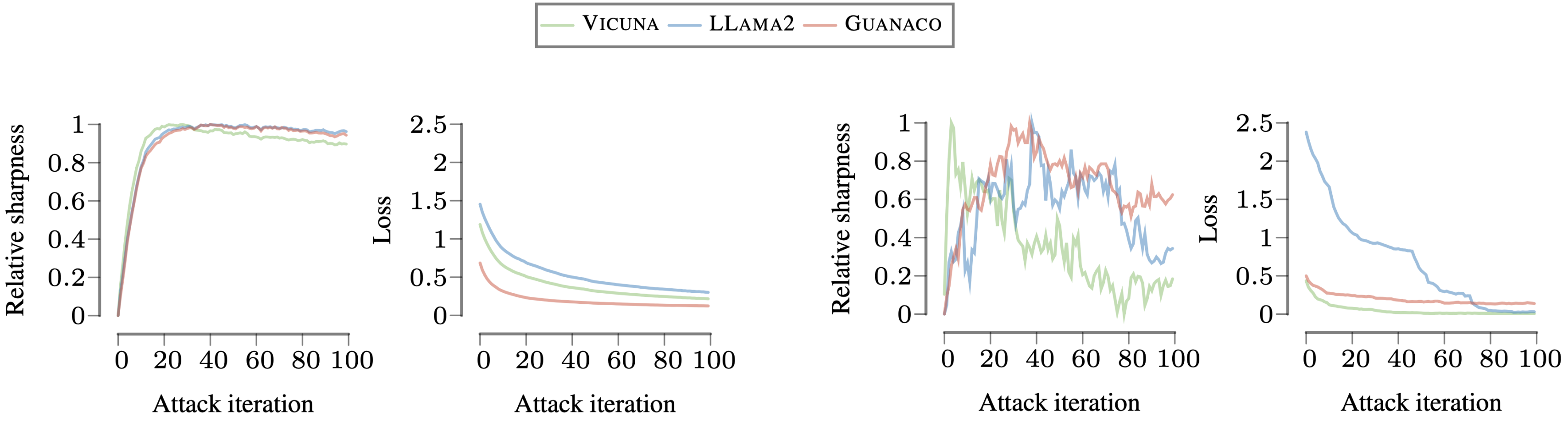
(a) Sharpness CIFAR-10

(b) Loss CIFAR-10

(c) Sharpness CIFAR-100

(d) Loss CIFAR-100

… even in LLMs



General trend

Examples of specific samples

Are we running in circles?

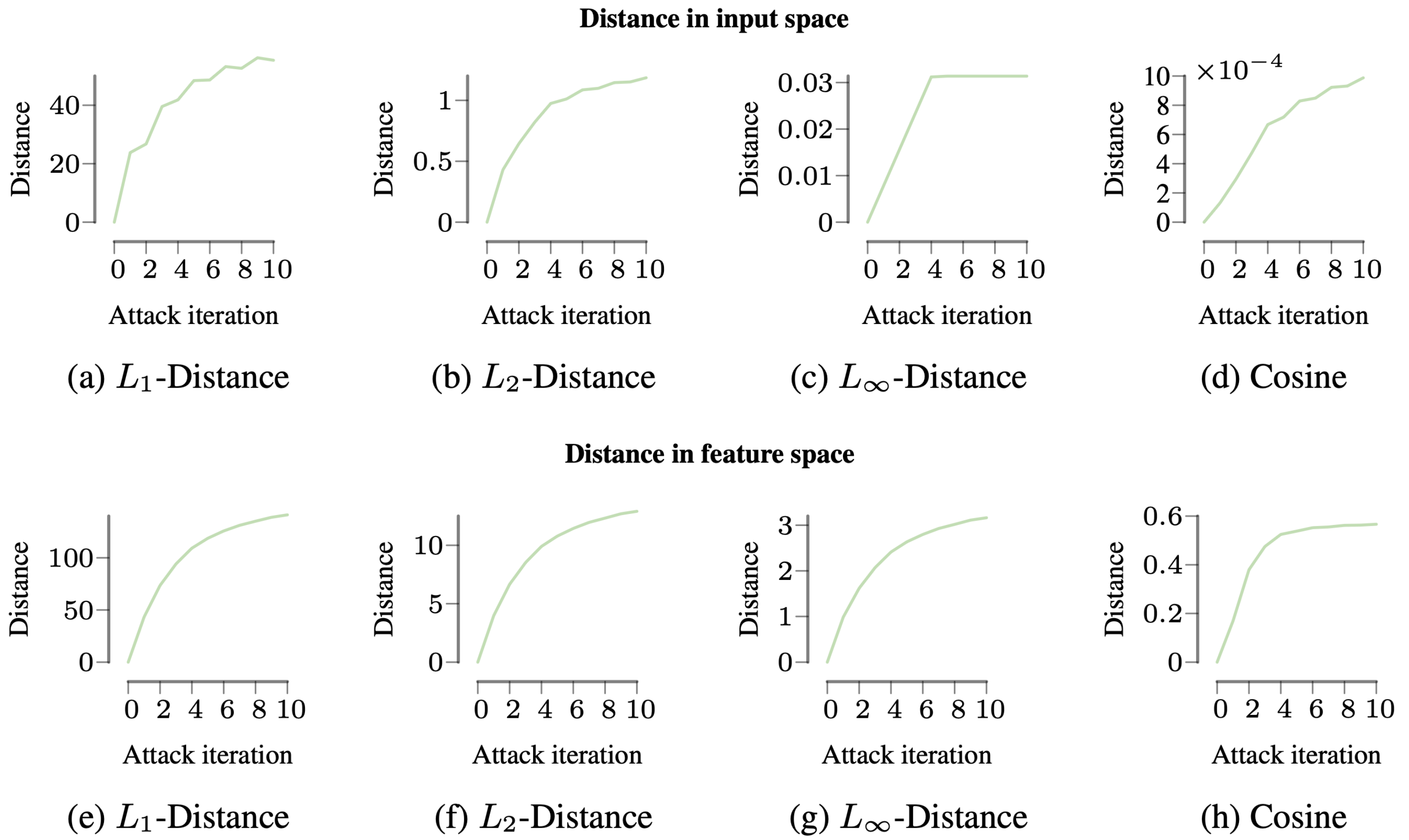


Figure 2: We show how far adversarial examples move in image/feature space from the initial image during a PGD-attack; we measure distance with L_1, L_2, L_∞ and cosine dissimilarity i.e. 1 - cosine similarity. We used CIFAR-10, WIDERESNET-28-4, and PGD with 10 iterations and δ=8/255.

Same geometry in earlier layers

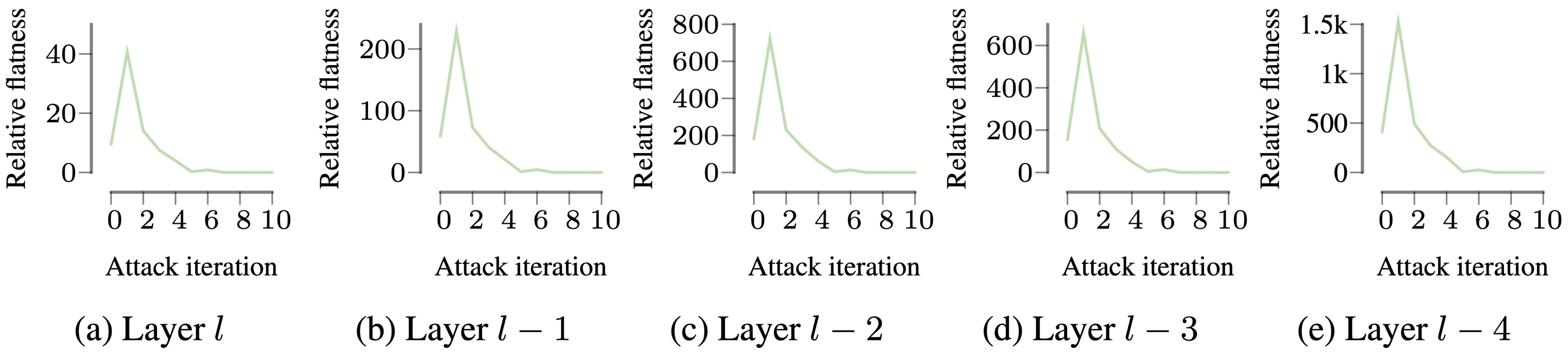
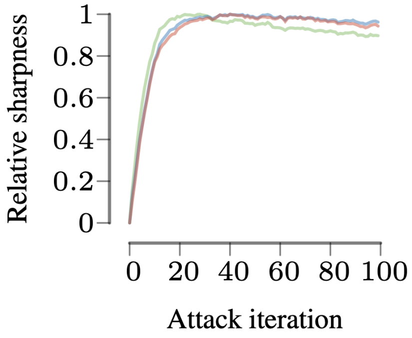


Figure 8: We show the relative sharpness measure computed in the penultimate layer l and in shallower layers $l-1$ to $l-4$ for WideResNet-28-4. Due to memory and runtime constraints, we approximate the measure using Hutchinson trace estimation used in Petzka et al. (2021) on 500 images. We observe the same phenomena as in the penultimate layer, which justifies that we focus only on the penultimate layer for our theoretical and experimental analysis.

Is it actionable?

Detecting adversarial attacks using flatness



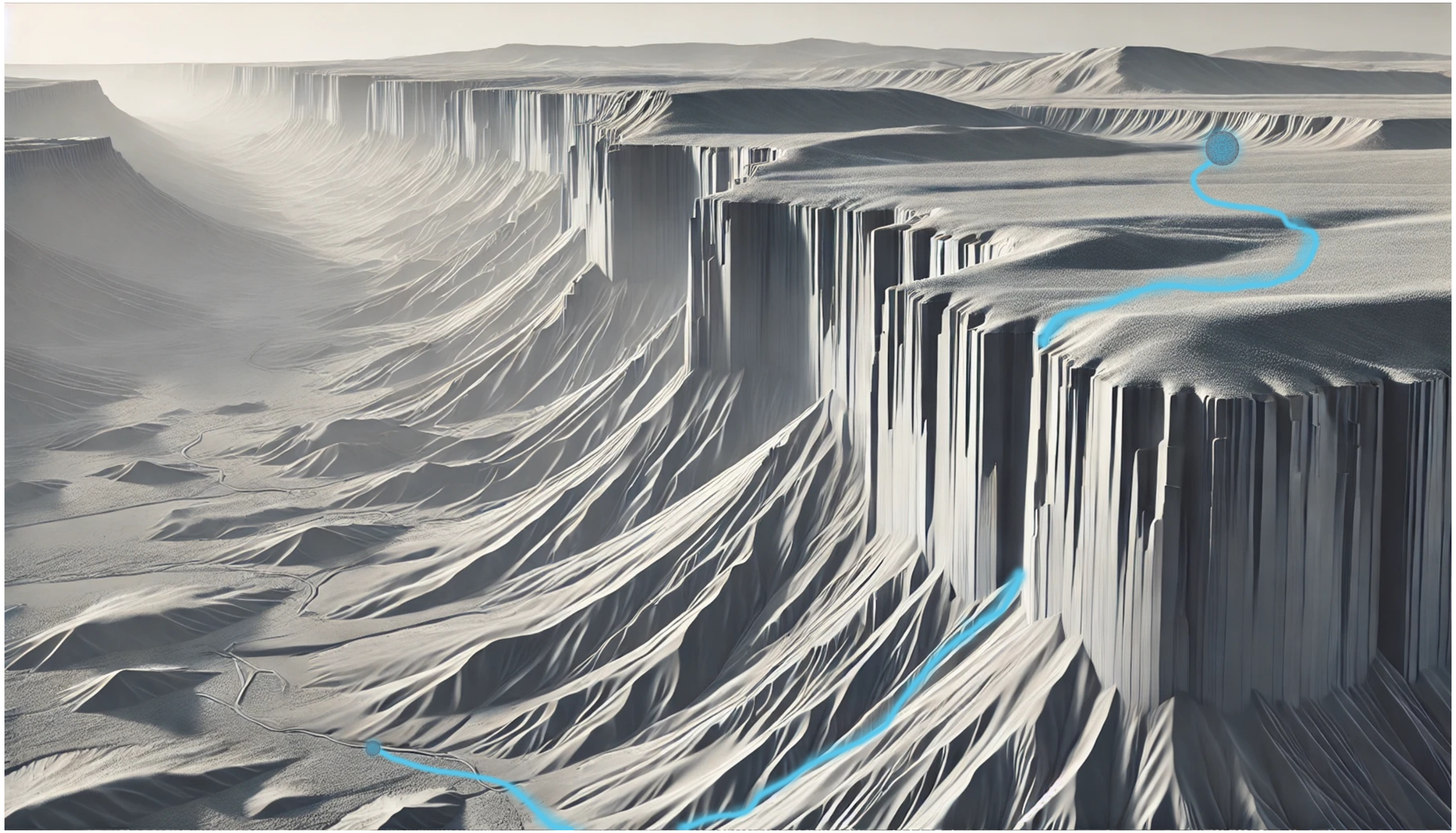
- Trace of the hessian $tr(H) = \sum_{j=1}^{k} \hat{y}_j(1-\hat{y}_j) \sum_{i=1}^{d} \phi_i^2$
- Extract a feature vector $\psi \in \mathbb{R}^k$

$$\psi_j = \hat{y}_j(1-\hat{y}_j) \sum_{i=1}^{d} \phi_i^2 \text{ , for } j = 1 \ldots k.$$

- Generate dataset by crafting adversarial examples (50:50)
- Train a Decision Tree on the features → How we found the uncanny valley
- Detects adversarial examples with 95% Accuracy

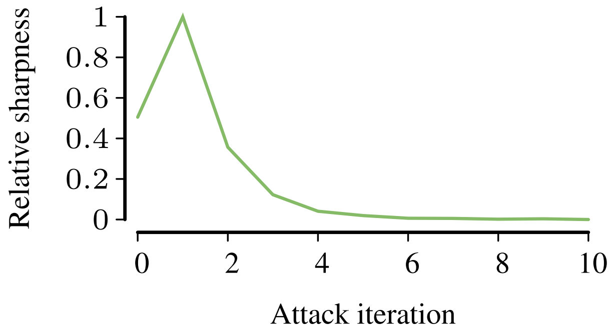
Intuition I

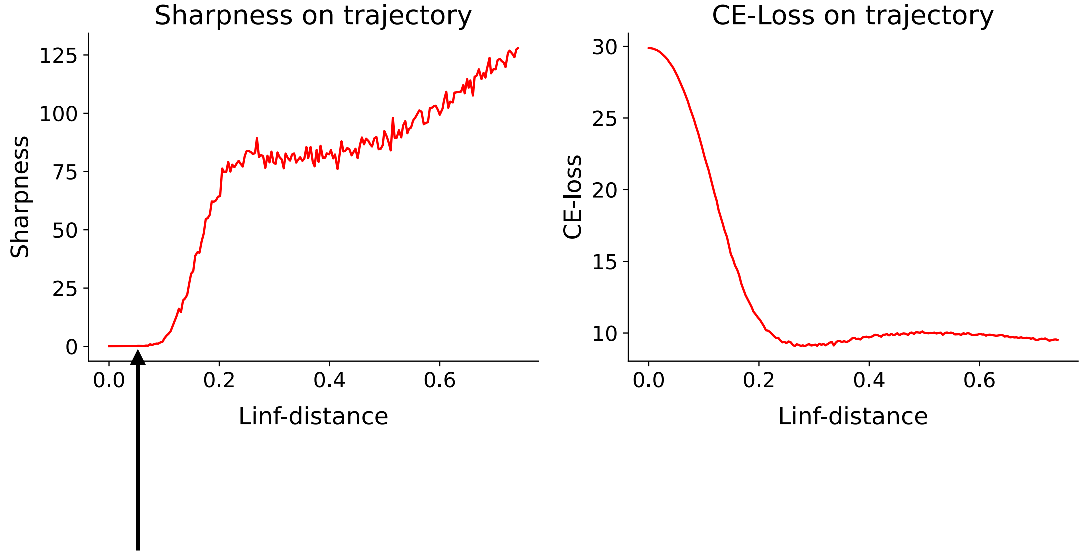
Broad Uncanny Valley



Intuition – Broad Uncanny Valley

Sample in the Uncanny Valley



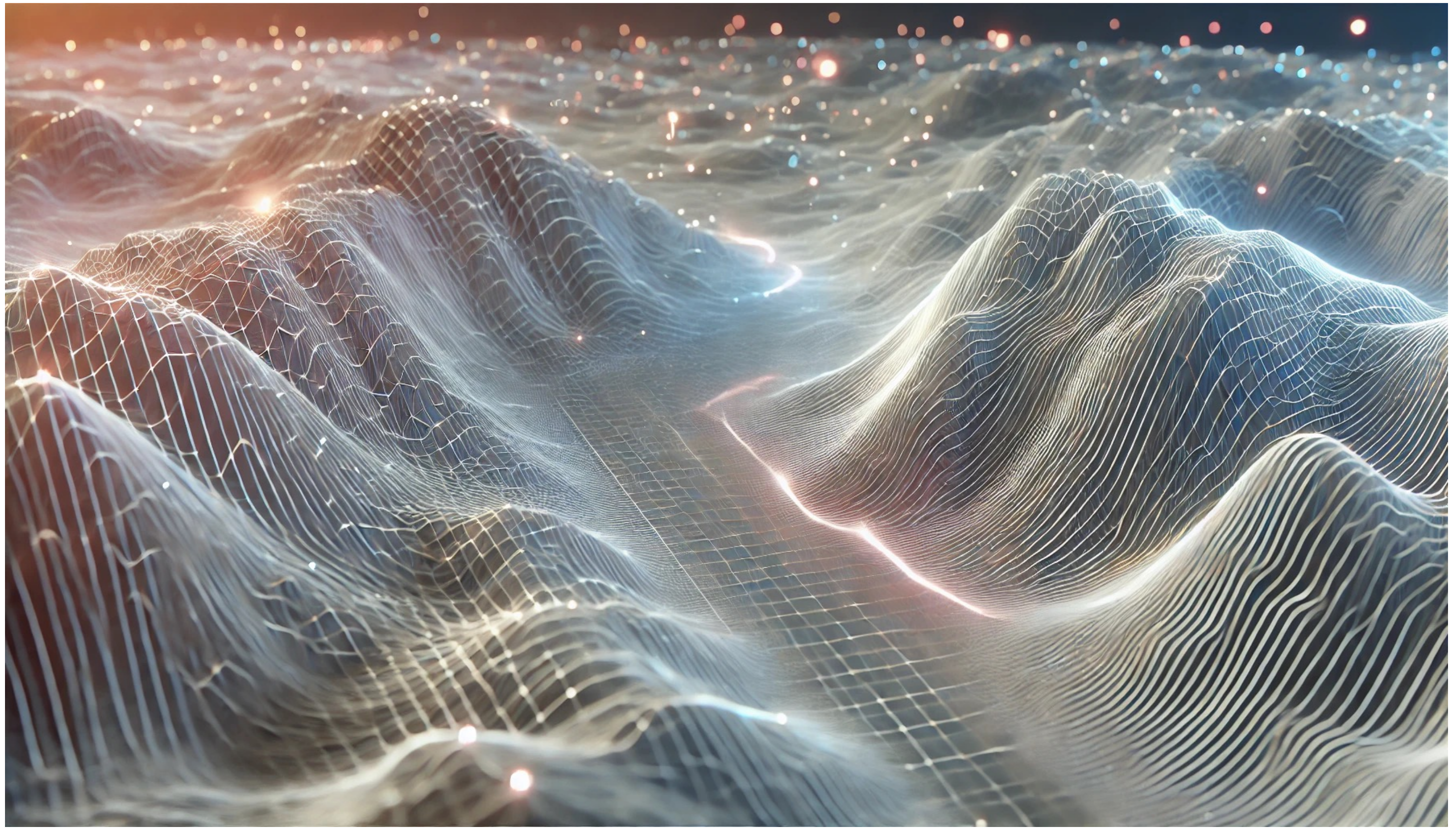


Not exactly zero i.e. ~0.06

- Generate adversarial examples
- Sample in the neighborhood using $\mathcal{N}(x_{adv}, \eta)$
- η chosen such that distance is limited

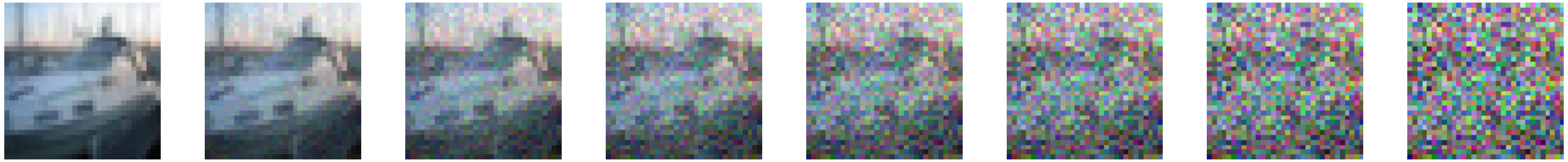
Intuition II

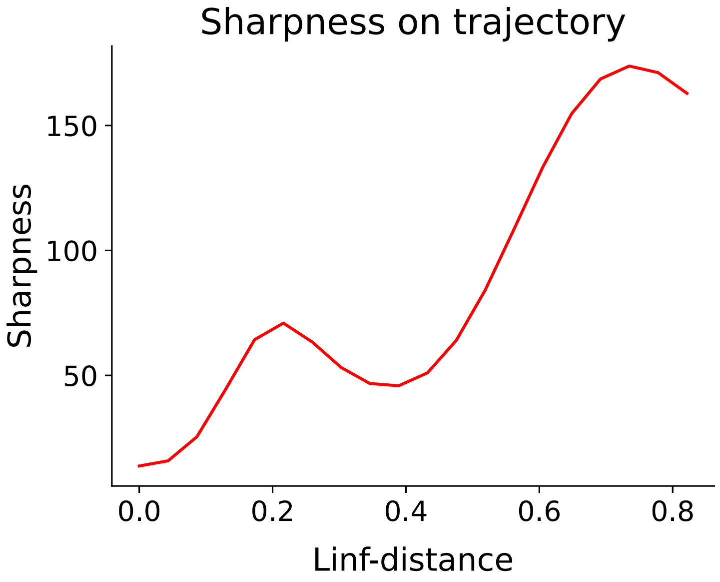
Wormholes to big blind spots

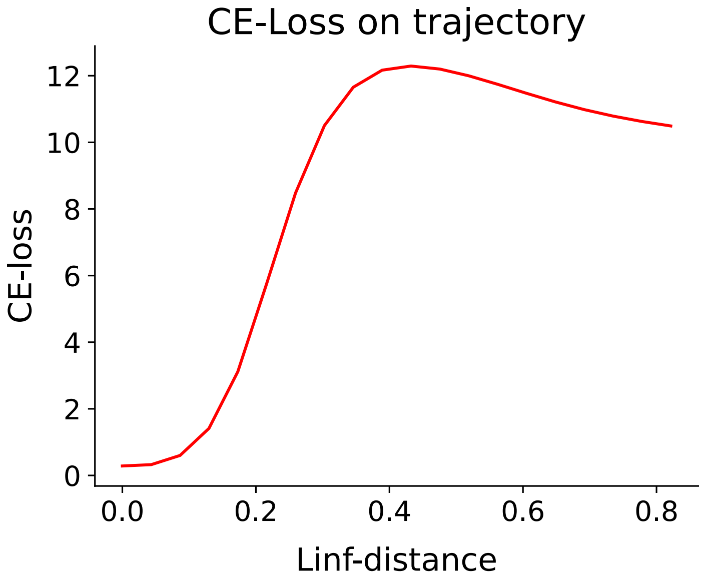


Intuition – Wormholes

Noise interpolation







Intuition – Wormholes

What we want:

1) Walk on the manifold
2) Walk to the blindspots

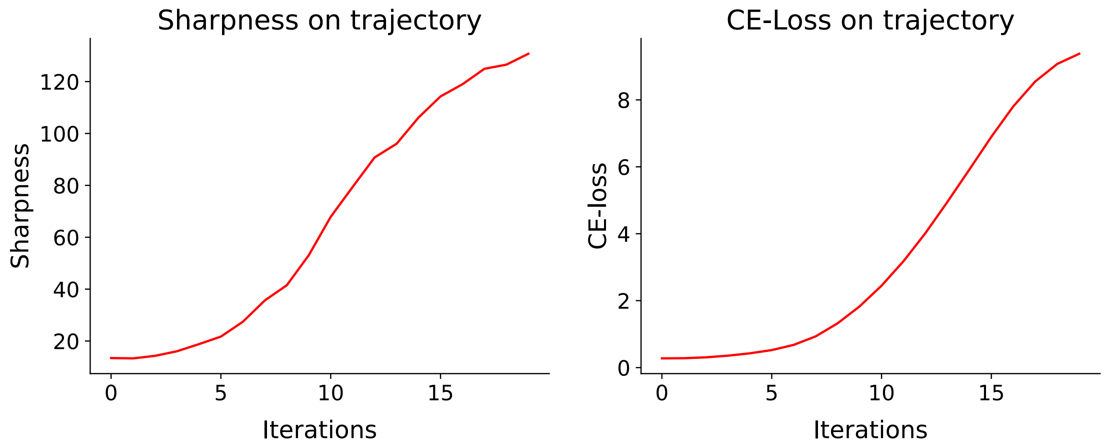
Idea: Compute the *span* and *kernel* of the data

Problem: Impossible to compute

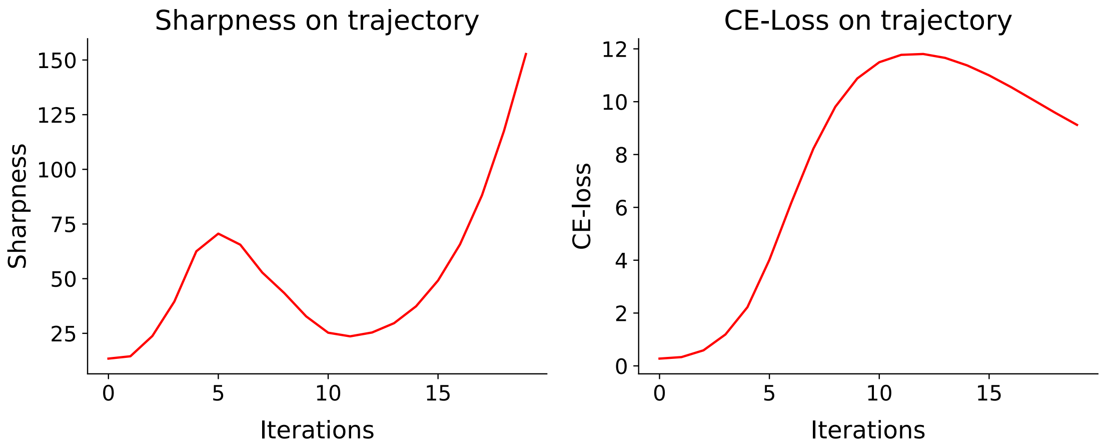
→ SVD and split according to singular values

→ Fix one point prop. to the SVs

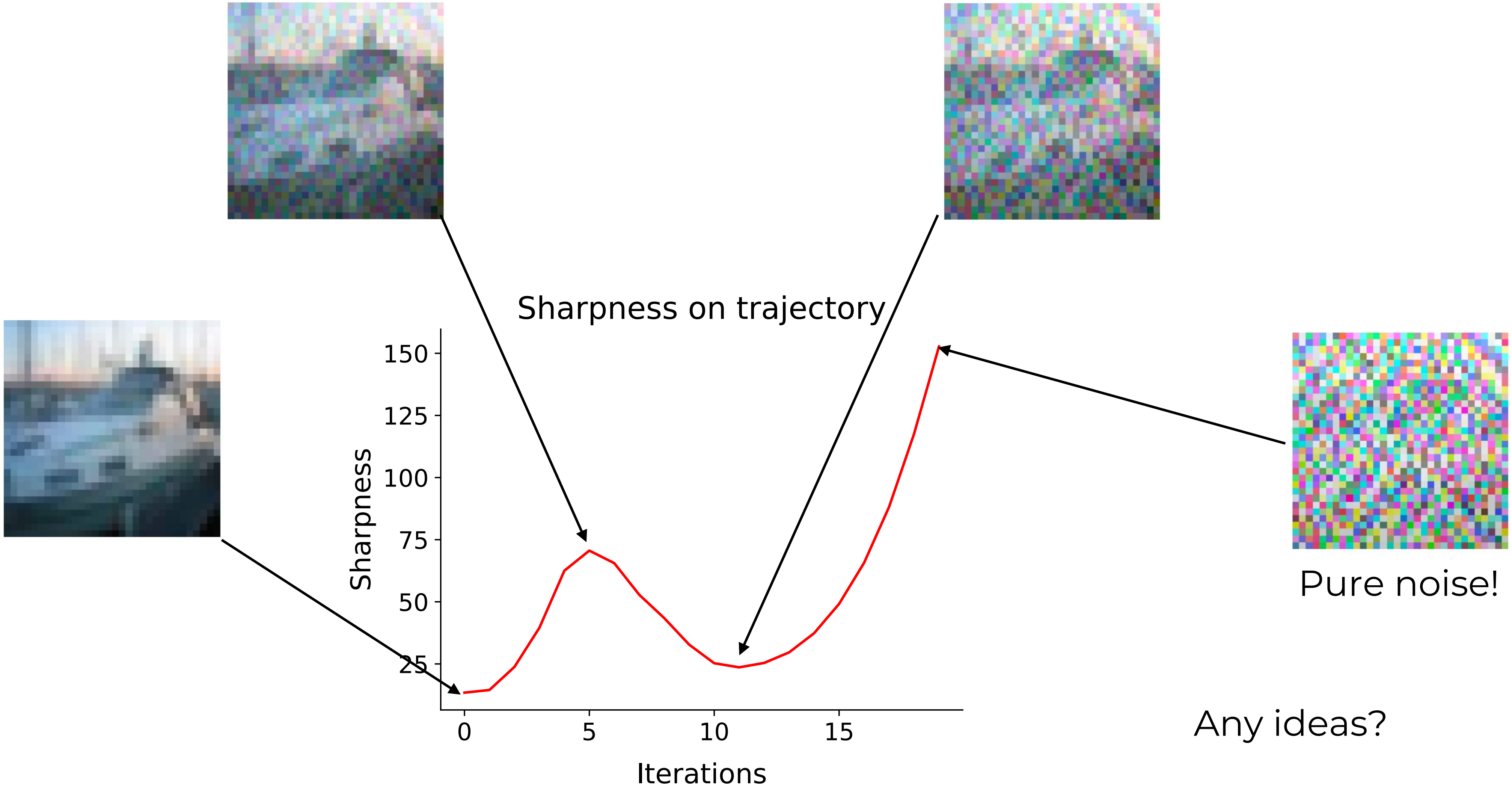
Manifold directions



Blindspot directions



Intuition – Wormholes

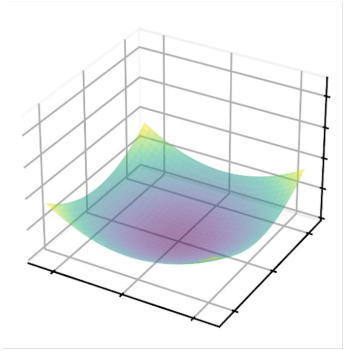


Any ideas?

Conlusion

Relative flatness:

$$K_{Tr}^{\phi} = \|\mathbf{w}\|_2 Tr(H)$$



Bound

$$\delta \propto \frac{\epsilon^{\frac{1}{3}}}{(L^3 k_{Tr}(\mathbf{w}))^{\frac{1}{3}}} + \frac{rkmL^2}{k_{Tr}(\mathbf{w})}$$