Some experiments and other remarks

Setup

13 Models

Name	Hyperparameters
Baseline 1	$\varnothing$
Baseline 2	Different seed
Jacobian 0.1	$\epsilon = 0.1$, $\eta = 0.03$
Jacobian 4.2	$\epsilon = 4.2$, $\eta = 0.03$
Jacobian 8.4	$\epsilon = 8.4$, $\eta = 0.03$
Isometry 0.1	$\epsilon = 0.1$, $\eta = 10^{-5}$
Isometry 4.2	$\epsilon = 4.2$, $\eta = 10^{-5}$
Isometry 8.4	$\epsilon = 8.4$, $\eta = 10^{-5}$
Adversarial Training	PGD, L_∞, step size=0.01, iter=40
Distillation	From baseline 4 (epoch 10), temp=20
Suppress max eigenvalue	$\eta = 0.1$
Parseval	$\varnothing$
Jacobian regularization only	$\eta = 0.03$

Setup

Shared hyperparameters

Dataset	MNIST
Epochs	10
Batch size	128
Optimizer	Adam, lr=0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$

Architecture: LeNet
Intput: $1 \times 28 \times 28$

Layer	Output dimension
Conv 3×3	$32 \times 26 \times 26$
Conv 3×3	$64 \times 24 \times 24$
MaxPool 2×2	$64 \times 12 \times 12 = 9216$
Linear + ReLU	128
Lienar	10

Training

9 plots

1. Training loss per batch*
2. Training cross entropy per batch*
3. Jacobian regularization $ReLU\left(\|\tilde{J}_x\|_2 - \frac{\delta(x)}{\rho(x)\epsilon}\right)$ per batch* (or isometry regularization for isometry models)
4. Spectral norm and Hölder upper bound per batch*
5. Frobenius norm per batch*
6. Bound $\frac{\delta(x)}{\rho(x)\epsilon}$ per batch*
7. Bound minus spectral norm per batch (no moving average)
8. Test loss per epoch
9. Test cross entropy per epoch

→ See the two other presentations

* with moving average over 50 batches

Remarks

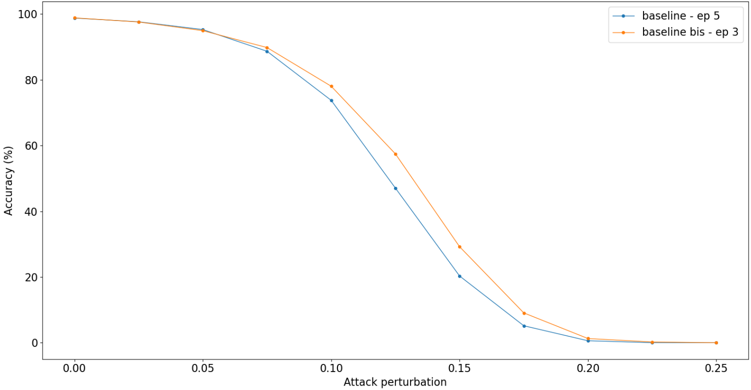
- The plots of baseline 2 are similar to the plots of baseline 1.
- The plots of Jacobian 4.2 are similar to the plots of Jacobian 8.4, except that there is no change of behavior at the end of training.
- The Frobenius norm of distillation seems smaller than its Spectral norm (which is impossible? Maybe an artifact of averaging).
- Hölder inequality is a good upper bound for the spectral norm but Frobenius norm is not.

Robustness testing

Projected Gradient Descent

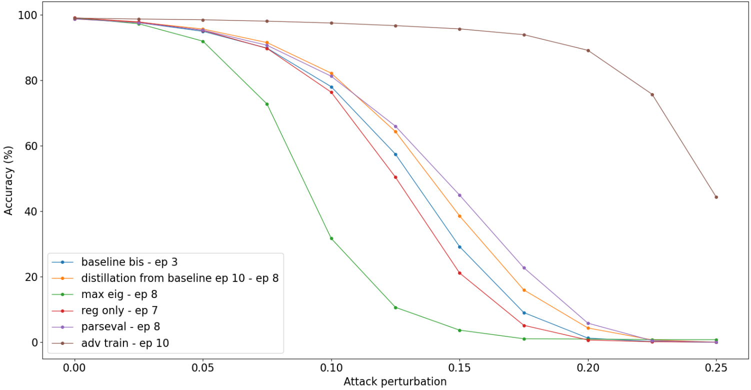
Norm	L_∞
Step size	0.01
Iterations	40
Random start	Yes

Robustness testing - Baseline alone



Baseline bis is best so we keep this one for the following plots.

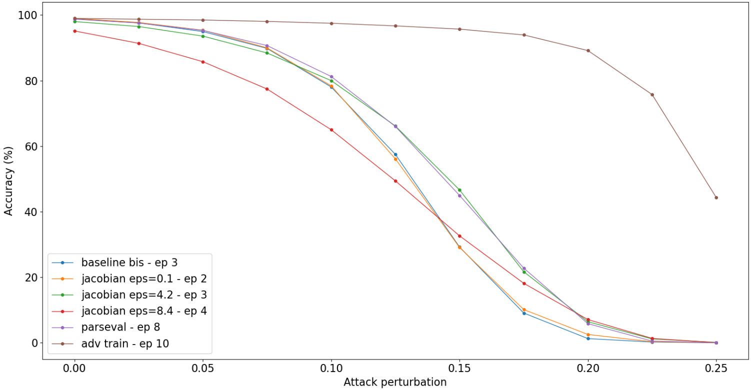
Robustness testing - All other defenses



Except *adversarial training*, *Parseval* is the best (it is outperformed by *distillation* for small perturbations).

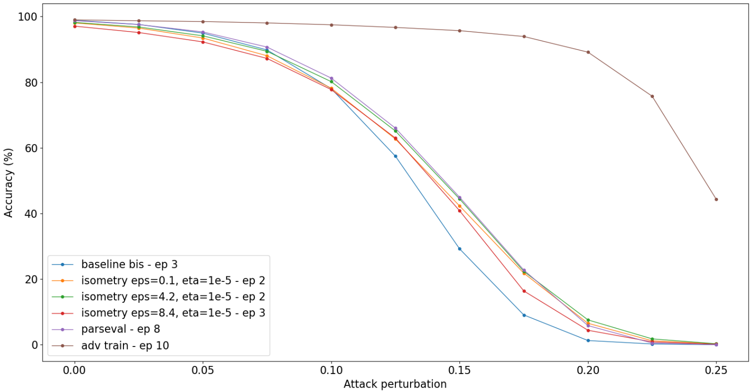
For some reason, *suppress max eigenvalue* is even worse than *baseline*. I trained it with $\eta = 0.02$ and $\eta = 0.1$ but the results are similar.

Robustness testing - Jacobian regularization



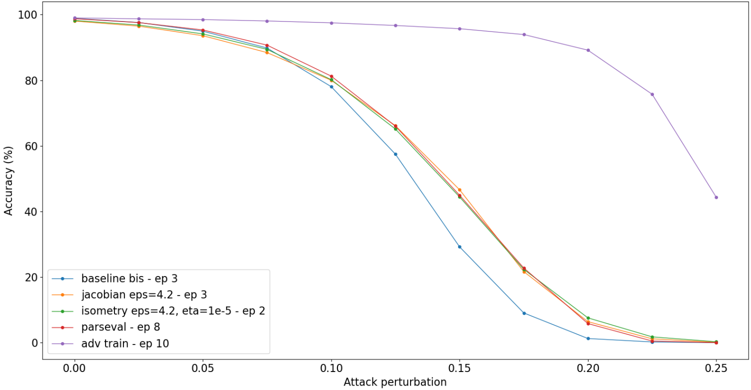
Jacobian regularization with $\epsilon = 4.2$ outperforms *parseval* for medium perturbations. I chose the weights of the epoch for which the regularization term is the lowest.

Robustness testing - Isometry regularization



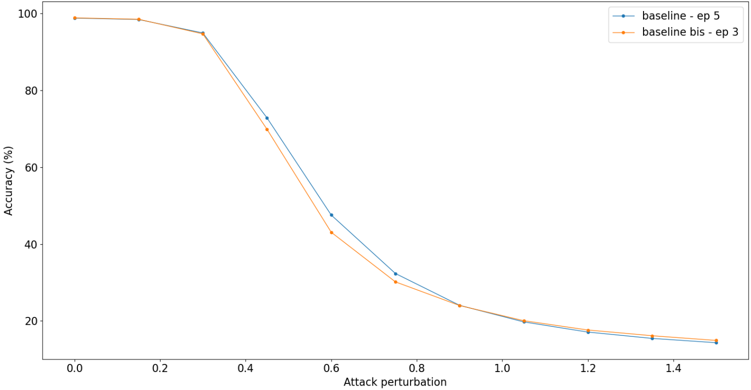
Isometry regularization with $\epsilon = 4.2$ is very close to *parseval*.

Robustness testing - Jacobian and isometry regularizations comparison



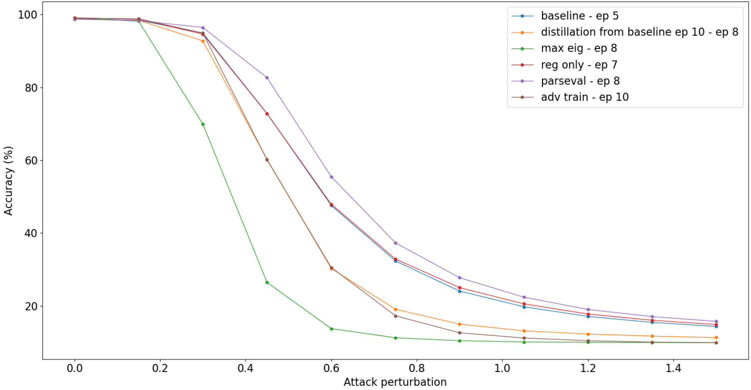
Jacobian regularization, *isometry regularization*, and *parseval* are very close to each other.

Robustness testing - Gaussian Noise - Baseline alone



Baseline is the best.

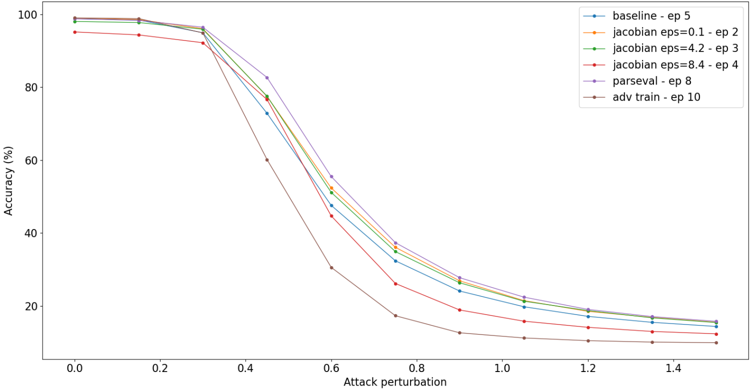
Robustness testing - Gaussian Noise - All other defenses



Parseval is the best.

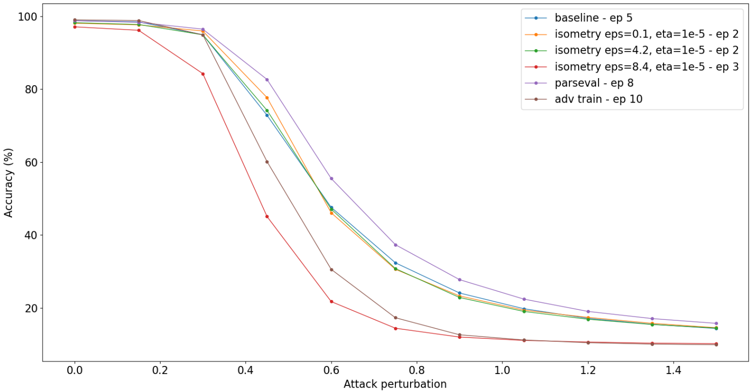
For some reason, *adversarial training* is worse than baseline against Gaussian noise.

Robustness testing - Gaussian Noise - Jacobian regularization



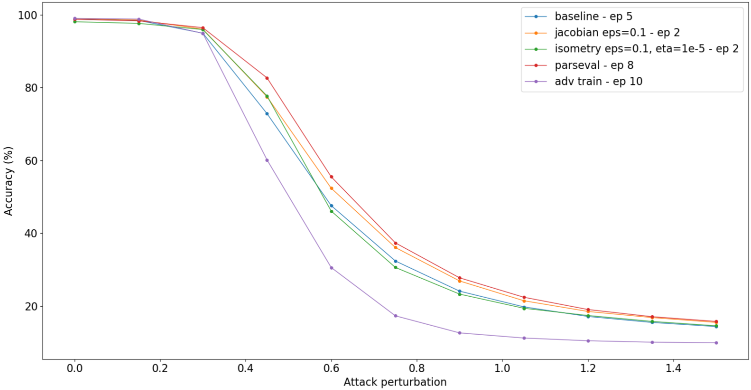
$\epsilon = 0.1$ is the best among *Jacobian regularization*, but very close to $\epsilon = 4.2$.

Robustness testing - Gaussian Noise - Isometry regularization



$\epsilon = 0.1$ is the best among *Isometry regularization*, but very close to $\epsilon = 4.2$.

Robustness testing - Gaussian Noise - Jacobian and isometry regularizations comparison



Parseval is the best, followed by *Jacobian regularization*.

Robustness testing

PGD budget: 0.15

AutoAttack (AA): Croce and Hein, 2020

- L_∞ budget: 0.15
- L_2 budget: 1.5

GN standard deviation: 0.75

Defense	Natural	PGD	AA L_∞	AA L_2	GN
Baseline 1	98.77	20.31	14.09	38.03	**32.34**
Baseline 2	**98.84**	**29.22**	**22.58**	**47.10**	30.16
Jacobian 0.1	**98.96**	29.08	23.49	45.45	**36.04**
Jacobian 4.2	98.01	**46.62**	39.92	49.13	34.87
Isometry 0.1	98.06	42.30	38.34	46.94	30.56
Isometry 4.2	98.21	44.43	**40.10**	**52.34**	30.75
Adv. Training	**98.98**	**95.69**	**95.43**	**73.34**	17.27
Distillation	98.69	38.60	8.84	30.25	19.05
Parseval	98.73	44.95	38.71	61.31	**37.31**

Plan for other experiments

- Other defenses: TRADES and FIRE (variants of adversarial training).
- Other dataset:
  - More complicated: CIFAR-10, Tiny ImageNet ...
  - More simple: e.g., a 2d linearly separable toy dataset to check that the method is sound and for easy visualization
  - For intuition and visualization: a logistic regression as in Picot et al., 2022
- Use a polytope closest point algorithm to compute δ (in Euclidean distance, and maybe Riemannian distance).
- Improve the computation of the Jacobian.
  - Approximate the matrix itself: Hoffman et al. 2019 seems efficient, see also Shafahi et al. 2019?
  - Approximations of the first eigenvalue / first singular value. Is it necessary since Hölder's inequality is already good?
- Do multiple runs for each model and report the runtime.

Even more potential experiments

- Evaluate the robustness with C&W?
- Is the code correct?
  Check that $\tilde{g}_x(X,X) \leq \frac{\delta(x)^2}{\epsilon^2}\overline{g}_x(X,X)$ whenever $\|\tilde{J}_x\|_2 \leq \frac{\delta(x)}{\epsilon\rho(x)}$.
- Visualization of Jacobian norm for all models (using spectral norm and L_∞ norm).
- ...

Plan for 2023

- January 18th → Submit the proposal.
- End of February → Submit the journal paper on Jacobian regularization.
- End of July → Certified defense using geometry is done (at least "theoretically").
- End of August → Paper with application of Jacobian regularization and/or certified defense to aviation is ready to be submitted. For example: defense against adversarial attack in aircraft trajectory prediction (see Tan et al. 2022).
- End of November → The thesis is written.

“Theoretical" remarks

1. Using “curvature" to derive the exact robustness criterion at a point $x \in \mathcal{X}$.
2. Is the Fisher metric the relevant metric for adversarial robustness?
3. Back to “first principles" for geometric-inspired certification method.

1 - Exact robustness criterion at a point

Reminders

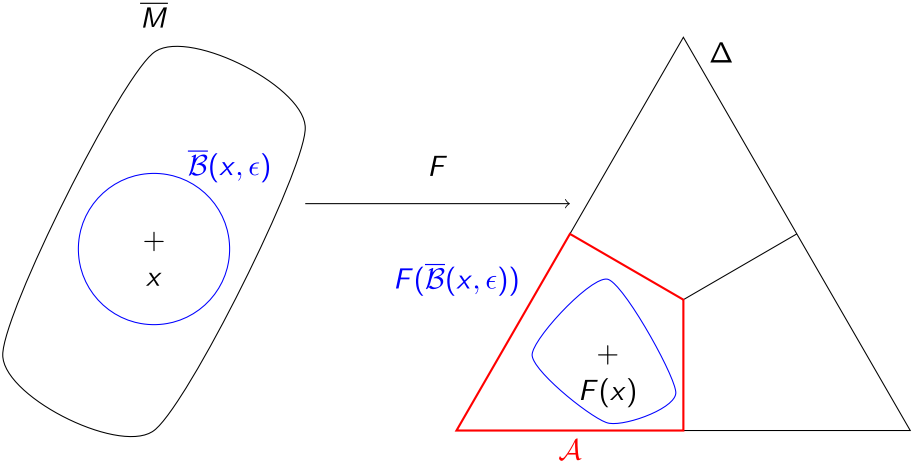
- $\mathcal{X}$ is a d-dimensional embedded submanifold of $\mathbb{R}^d$.
- Δ is the $(c-1)$-simplex.
- g is the Fisher metric on Δ.
- $F : \mathcal{X} \to \Delta$ is a smooth map.

We are interested in two Riemannian structures on $\mathcal{X}$:

- $\overline{M} = (\mathcal{X}, \overline{g})$ where $\overline{g}$ is the Euclidean metric.
- $M = (\mathcal{X}, \tilde{g})$ where $\tilde{g} = F^* g$ is the pullback metric of g by F.

Now, let $x \in \mathcal{X}$ and let $\epsilon > 0$.
We want to find a criterion on F such that F is robust to any L_2 attack at x with a budget less than ϵ.

1 - Exact robustness criterion at a point



1 - Exact robustness criterion at a point

- Let $\overline{\mathcal{B}}(x, \epsilon) = \{z \in \mathcal{X} : \|z - x\|_2 < \epsilon\}$.
- Let $\mathcal{A} = \{\theta \in \Delta : \arg\max \theta = \arg\max F(x)\}$ be the set of points in Δ with the same class as x.

A **complete** criterion should ensure that $\boxed{F(\overline{\mathcal{B}}(x, \epsilon)) \subseteq \mathcal{A}}$.

However, $\mathcal{A}$ is too complicated[†].

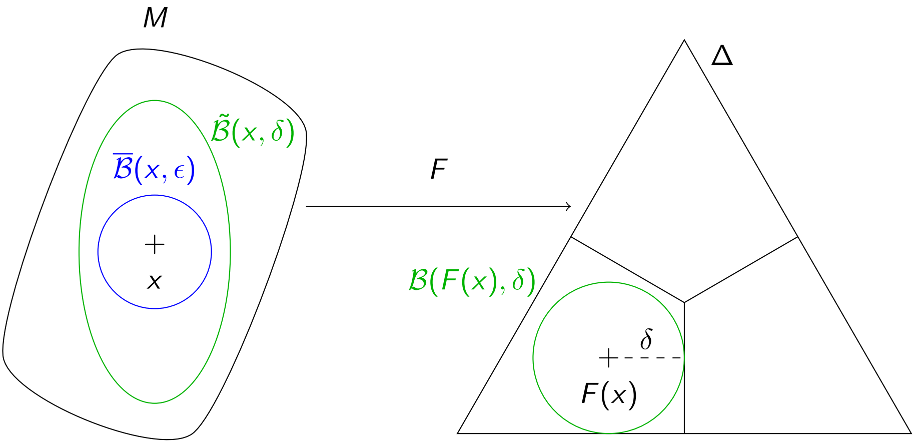
Thus, we will be looking for a **sound** but incomplete criterion

$\rightarrow$ the criterion is still **exact** because any L_2 attack with a budget less than ϵ will fail, but there may exist points for larger budgets that are still robust

$\rightarrow$ the criterion is "too strong".

[†]For now ... cf. slide 31 and after

1 - Exact robustness criterion at a point



1 - Exact robustness criterion at a point

- Let δ be the Riemannian distance between $F(x)$ and the decision boundary (or any approximation of it).
- Let $\mathcal{B}(F(x), \delta) = \{\theta \in \Delta : \theta = \exp_{F(x)}(v), g_{F(x)}(v, v) < \delta^2\}$ be a geodesic ball.
- Similarly, let $\tilde{\mathcal{B}}(x, \delta) = \{z \in \mathcal{X} : z = \exp_x(v), \tilde{g}_x(v, v) < \delta^2\}$
- By definition, $F(\tilde{\mathcal{B}}(x, \delta)) = \mathcal{B}(F(x), \delta)$.
- Note that $\overline{\mathcal{B}}(x, \epsilon) = \{z \in \mathcal{X} : z = \overline{\exp}_x(v), \overline{g}_x(v, v) < \epsilon^2\}$

<u>Assumption</u>[‡]: $\mathcal{A} \approx \mathcal{B}(F(x), \delta)$.

The sound, exact, but incomplete criterion is: $\boxed{\overline{\mathcal{B}}(x, \epsilon) \subseteq \tilde{\mathcal{B}}(x, \delta)}$.

[‡]cf. slide 31 and after.

1 - Exact robustness criterion at a point

What is exp**?**
Let $x = (x^1, \ldots, x^d)$ in the standard coordinates of $\mathbb{R}^d$.
The curve $\gamma(t) = (\gamma^1(t), \ldots, \gamma^d(t))$ is the geodesic starting at x with initial velocity $v = (v^1, \ldots, v^d)$ if for all $k \in \{1, \ldots, d\}$:

$$\begin{cases} \gamma^k(0) = x^k \\ \frac{d\gamma^k}{dt}(0) = v^k \\ \frac{d^2\gamma^k}{dt^2}(t) + \sum_{i=1}^d \sum_{j=1}^d \frac{d\gamma^i}{dt}(t)\frac{d\gamma^j}{dt}(t)\Gamma_{ij}^k(\gamma(t)) = 0 \end{cases} \tag{1}$$

If $\gamma(t)$ is the solution of the initial value problem (1), then we define $\exp_x(v) = \gamma(1)$.

1 - Exact robustness criterion at a point

The "curvature" is given by the Christoffel symbols Γ_{ij}^k.
Let $(r^1, \ldots, r^d)$ be the standard coordinates of $\mathbb{R}^d$. Then:

$$\Gamma_{ij}^k(x) = \frac{1}{2} \sum_{l=1}^{d} \left(\tilde{g}^{-1}\right)_x^{kl} \left(\frac{\partial \tilde{g}_{x,jl}}{\partial r^i} + \frac{\partial \tilde{g}_{x,il}}{\partial r^j} - \frac{\partial \tilde{g}_{x,ij}}{\partial r^l} \right). \tag{2}$$

Remember that in coordinates, the matrix of $\tilde{g}_x$ is $\tilde{G}_x = J_x^T G_{F(x)} J_x$.
Thus, to compute Γ_{ij}^k at x, we need to compute:

- The derivative $J_x^T G_{F(x)} J_x$ with respect to x. Maybe, the Hessian matrix of F appears hear?
- The inverse of $J_x^T G_{F(x)} J_x$.

And then, we need to solve the geodesic equation (1) ...

1 - Exact robustness criterion at a point

- How to efficiently solve the geodesic equation (1) seems to be an extensively studied problem, but I don't know much about it for now ...
- There are also methods called "retractions" that aim at approximating the exponential map (exp).
- Is it possible to take advantage of the structure of F (it's a neural network) to obtain a simpler expression for exp?

1 - Exact robustness criterion at a point

Assume that we have a procedure to efficiently approximate $z = \exp_x(v)$.
In fact, we are more interested in the inverse map $v = \log_x(z)$.
In coordinates, the criterion becomes:

$$z^T z < \epsilon^2 \Rightarrow \log_x(z)^T \tilde{G}_x \log_x(z) < \delta^2. \tag{3}$$

We can also write the criterion as:

$$\max_{v \in \log_x\left(\overline{\mathcal{B}}(x,\epsilon)\right)} v^T \tilde{G}_x v < \delta^2. \tag{4}$$

2 - Is the Fisher metric the relevant metric for adversarial robustness?

- As we can see in equation (4), the choice of the metric g on Δ is of great importance.
- We chose the Fisher metric because it is supposed to have good properties, but in fact it is mainly by tradition.
- The Fisher metric has good properties if we see Δ as the family of categorical distributions. But do we really care?

2 - Is the Fisher metric the relevant metric for adversarial robustness?

- Concerning the problem of adversarial robustness, the sole and only important property is: **is there a ball of g centered on $F(x)$ that is a good approximation of $\mathcal{A}$?** If g is the Fisher metric, the answer is NO!
- If we want to apply the method described above, and if we want to have a criterion as complete as possible, then we can see that **the right metric $g^{(F(x))}$ depends on $F(x)$.**

$\rightarrow$ **Given $F(x)$, how can we efficiently find a suitable $g^{(F(x))}$?**

3 - Certifiable defense: back to first principles

- My intuition tells me that adversarial robustness is a **generalization** issue.
- Instead of "understanding the real nature of the task", current machine learning models are looking for spurious correlations that works well on most training and test examples but are fundamentally flawed, thus the existence of adversarial examples.

3 - Certifiable defense: back to first principles

Learning can be split into three components:

1. **The training loss**: what we minimizes must correspond to what we want to do.
   $\rightarrow$ *Consistency and calibration of adversarial surrogate losses.*
2. **The training algorithm**: the type of minimum we are converging to impacts the generalization.
   $\rightarrow$ *PAC-Bayes, stochastic optimization: sharp vs flat minima, saddle points (global minima with rank constraint) ...*
3. **The model architecture**: neural networks are not black-box.
   $\rightarrow$ *Approximation theory (?): VC dimension, Rademacher complexity, fat-shattering dimension ...*